2020

Action Recognition in Still Images: Confluence of Multilinear Methods and Deep Learning

Marjaneh Safaei
University of Central Florida

ACTION RECOGNITION IN STILL IMAGES: CONFLUENCE OF MULTILINEAR METHODS AND DEEP LEARNING

by

MARJANEH SAFAEI
M.S. Eastern Mediterranean University, 2010

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2020

Major Professor: Hassan Foroosh

ABSTRACT

Motion is a missing information in an image, however, it is a valuable cue for action recognition. Thus, lack of motion information in a single image makes action recognition for still images inherently a very challenging problem in computer vision. In this dissertation, we show that both *spatial* and *temporal* patterns provide crucial information for recognizing human actions. Therefore, action recognition depends not only on the spatially-salient pixels, but also on the temporal patterns of those pixels. To address the challenge caused by the absence of temporal information in a single image, we introduce five effective action classification methodologies along with a new still image action recognition dataset. These include (1) proposing a new Spatial-Temporal Convolutional Neural Network, *STCNN*, trained by fine-tuning a CNN model, pre-trained on appearance-based classification only, over a novel latent space-time domain, named *Ranked Saliency Map and Predicted Optical Flow*, or $Rank_{\text{SM-POF}}$ for short, (2) introducing a novel unsupervised Zero-shot approach based on low-rank *Tensor Decomposition*, named *ZTD*, (3) proposing the concept of *temporal image*, a compact representation of hypothetical sequence of images and then using it to design a new hierarchical deep learning network, *TICNN*, for still image action recognition, (4) introducing a dataset for STill image Action Recognition (*STAR*), containing over $1M$ images across 50 different human body-motion action categories. *UCF-STAR* is the largest dataset in the literature for action recognition in still images, exposing the intrinsic difficulty of action recognition through its realistic scene and action complexity. Moreover, *TSSTN*, a two-stream spatiotemporal network, is introduced to model the latent temporal information in a single image, and using it as prior knowledge in a two-stream deep network, (5) proposing a parallel heterogeneous meta-learning method to combine *STCNN* and *ZTD* through a stacking approach into an ensemble classifier of the proposed heterogeneous base classifiers. Altogether, this work demonstrates benefits of *UCF-STAR* as a large-scale still images dataset, and show the role of latent motion information in

recognizing human actions in still images by presenting approaches relying on predicting temporal information, yielding higher accuracy on widely-used datasets.

To my parents, Majid and Nahid

Thank you for your unconditional love, support, guidance and encouragement throughout my life.

I am proud to be your daughter.

LOVE YOU

ACKNOWLEDGMENTS

First and foremost, I would like to sincerely thank my doctoral advisor, Dr. Hassan Foroosh, for his consistent guidance, support and patience. His invaluable advises and encouragements have made my research journey possible. My sincere appreciation is extended to the members of my dissertation committee, Dr. Charles Hughes, Dr. Ulas Bagci and Dr. Valerie Sims for their constructive comments and contributions to a timely completion of this dissertation. I would also like to thank my dear labmate and best friend, Vildan, for being there for me when most needed, listening to me when I was much annoying and making me laugh all the time. My deepest gratitude is for my parents for giving me the strength to reach for the stars and chase my dreams by their endless love and support since day one. And last but not least, I would like to thank my husband Pooyan for all the encouragements, supports and his undeniable role in my PhD.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

This chapter provides an introduction to the area of human action recognition, action recognition in still images, and our main contributions in this area of research.

1.1 Human Action Recognition

Human action has different definitions in psychology, sociology and philosophy. In cognitive psychology, many theories suggest that the interactions between perception and action systems is where the important cognitive functions reside in. An action system can offer a way to predict the future consequences of currently perceived actions. Specifically, such predictions might deliver higher accuracy when an individual observes his/her own actions than when observes an action performed by another individual.The reason involved is that in the former case the system that plans the action is the same system that is utilized to predict the action's effects. The claim that perceptual input can be linked with the action system to predict future outcomes of actions is heavily supported by the evidence pointed out above.

In computer vision, human action focuses on the human external performance in daily life. Humans can understand the action and the purpose of the actor through human vision system. However, monitoring human actions in a variety of real-world scenarios using human labors is too expensive. Building a machine that can accurately understand humans actions and intentions is one of the ultimate goals of computer vision as a field of artificial intelligence research. Human action recognition has a wide range of applications, such as video storage and retrieval [1, 2], automated video surveillance [3, 4, 5], surveillance camera networks [6, 7, 8], image context analysis and annotation [9, 10], humanmachine interface [11, 12] and identity recognition [13].

Action detection [14, 15, 16] and action recognition [17, 18] are two basic topics in the computer vision community. Action detection involves locating actions of interest in space and/or time whereas action recognition refers to the act of classifying a human action that is present in a video containing complete action execution. Human action recognition infers the action label after observing the entire action execution. On the other hand, inferring action labels from incomplete video data [19] or a single image occurs when observing complete action execution is not an option. Predicting future state of a human action based on incomplete action executions is referred to as action prediction. Thus, action recognition in still images can be treated as an extreme case of the action prediction problem [20].

1.2 Action Recognition in Still Images

Which parts of the human body are likely to move? What are their overall shapes? and In which direction they may move? Answering these questions may be viewed as a non-semantic form of action prediction when dealing with still images. Action recognition in still images has important applications in areas such as image search and retrieval, image annotation, and video summarization, to name a few. Considering, the tremendous growth in the number of images on the Web, it is of paramount importance to automate the analysis of human actions in still images. Although, there has been a remarkable progress on human action recognition in video data, action recognition in still images remains more challenging and less attended by researchers. Recent methods on single image action recognition typically represent actions by spatial features. In this work, we argue for the importance of a latent space-time representation of action in a single image, derived from both spatial and latent temporal cues.

An obvious challenge in still image action recognition is the absence of motion information. While in videos one can readily infer motion [21, 22, 23, 24], such information is missing in a single im-

Figure 1.1: Still images depicting actions recognizable by humans.

age. The problem becomes exacerbated when there is no contextual information, *e.g*. when no objects (other than the human) are present in the image. Interestingly, humans are less challenged to perform this task compared to machines. Human brain is able to not only recognize image content but also predict what action might take place by a person in the scene, by predicting the most probable motions. As a recent study in applied perception shows [25], humans use motion prediction extensively to predict actions based on their past experience, especially when the recognition relies mainly on human body motions, and not human-object interaction. We interpret this ability of humans to predict and visualize the most likely motions of body parts in a still image

as an ability to infer a *latent space-time* representation domain. For instance, when humans see a running person in a still image they are able to imagine and visualize the motion, despite the fact that the single image does not contain any motion. This prompts us to argue that both spatial and the latent temporal patterns provide important information for recognizing human actions in still images. Fig. 1.1 depicts some actions recognizable by humans when exposed to still images.

This dissertation presents multiple significant contributions to this area, which have been previously published in conference proceedings including the AAAI Conference on Artificial Intelligence, the IEEE International Conference on Image Processing (ICIP), and the IEEE Winter Conference on Applications of Computer Vision (WACV). A portion of this dissertation proudly received the "Best Paper Award" at the 2018 IEEE International Conference on Image Processing. We aim to present these works within the unified framework of action recognition in still images. We hope that this dissertation will encourage readers to promote future research opportunities in this area.

1.3 Main Contributions

The **first** contribution presented in this work is introducing the latent space-time domain-mapping. We propose a method of explicitly learning the latent space-time domain, herein referred to as *Ranked Saliency Map and Predicted Optical Flow*, or $Rank_{\text{SM-POF}}$ for short. We use the predicted optical flow in a still image as the latent temporal information, while using the saliency map to represent the most significant spatial information. A Rank-SVM is then trained to project a given image onto the latent domain $Rank_{\text{SM-POF}}$, using both saliency map and the predicted optical flow. The result of the projection or mapping is an image forming a compact space-time representation that ranks/encodes each pixel in terms of its predicted motion and spatial saliency. Mapping to the latent space-time domain, makes it possible to model still image action recognition as a transfer

learning problem. Thus, by fine-tuning a CNN model that is pre-trained on appearance only, we obtain a new Spatial-Temporal Convolutional Neural Network (*STCNN*) model trained to classify actions in still images using their latent space-time representation. This work was presented at WACV 2019 [26].

The **second** contribution is proposing *ZDT*, a novel unsupervised Zero-shot still image action classification approach based on low-rank Tensor Decomposition. Here, we represent the latent space-time domain for each action class using a prototype group of images from that class. Therefore, the latent domain is implicitly modeled as a subspace spanned by the group, forming a 3-way tensor for each action class. Action classification for a test image is then achieved by identifying the prototype group whose low-rank representation defines a subspace that is closest to the test image, *i.e*. by measuring to what extent the test image would perturb such low-rank representation for each class, with the perturbation being minimum for the correct class. This work was presented at IEEE ICIP 2018 [27].

The **third** contribution is proposing a novel framework, TICNN, composed of two convolutional neural networks. The first CNN learns a novel model to predict non-existing sequence of images given an image I, summarizing them into a single image containing multiple frames of a hypothetical video, hereinafter referred to as *temporal image*. The second CNN extracts temporal image features to classify human actions in still images. To the best of our knowledge, this is the first attempt to predict temporal images, dynamic patterns of a still image, to alleviate the lack of motion information in still images. This work was presented at ICIP 2018 and received the best paper award [28].

The **fourth** contribution is two-fold: (i) introducing a new large-scale multi-modal dataset, *UCF-STAR*, to advance the current still image-based action recognition research through its additional rich and structured metadata. *UCF-STAR* contains 1,038,622 annotated still images, collected from

the *wild*, more than 40 times the size of the largest previous action image dataset;
ie BU-101 [29]. The key characteristics of *UCF-STAR* include (1) focusing on human body-motion rather than relatively static human-object interaction categories, (2) collecting images from the *wild* to benefit from a varied set of action representations, (3) appending multiple human-annotated labels per image rather than just the action label, and (4) inclusion of rich, structured and multi-modal set of metadata for each image. (ii) developing a two-stream spatiotemporal network (TSSTN), similar to networks used in the video literature, and decomposed still image action recognition into spatial and predicted temporal streams. This work was presented at AAAI 2020 [30].

The **fifth** contribution is proposing an ensemble classifier comprising of two very heterogeneous base classifiers, i.e. one highly dependent on training data for explicitly learning the latent space-time representations, and the other independent of training, relying solely on prototype groups of images that span the latent space-time domain for each action class. Due to their heterogeneity, the two base classifiers are complementary in terms of their per-class accuracy. Therefore, a *Meta-Classifier* is learned to combine the two *base-level* classifiers to tackle the challenges of still image action recognition.

Finally, the realistic complexity of *UCF-STAR* exposes the inherent difficulty of human body-motion action recognition, overlooked by many well-known significantly smaller datasets. We perform comparative benchmarking of well-known methods on Stanford-40 [31], Willow [32], WIDER [33], BU-101 [29] and *UCF-STAR*. Results confirm the more challenging nature of *UCF-STAR* compared to other datasets. An extensive experiments on UCF-*STAR* and several most popular datasets clearly demonstrate that appearance and the predicted latent motion are complementary sources of information, which together lead to significant performance improvement in still image action recognition, outperforming the state-of-the-art methods.

CHAPTER 2: LITERATURE REVIEW

We divide this chapter into three sections. First, we review the related works on single image action recognition. Next, we provide an overview of the ensemble methodologies. Finally, the existing datasets used as benchmarks in this area is further elaborated on..

2.1 Still Image Action Recognition

In the context of still image action recognition, a big challenge is the lack of temporal information, thus the traditional spatiotemporal features [34] cannot be applied. In conventional action recognition methods based on videos, the low-level features extracted from space-time prove effective in action recognition [35, 36, 37, 38]. In contrast, low-level features extracted from a single image do not work well in the realm of still image action recognition.

Human action recognition in still images has gained increasing attention in recent years [39, 40, 41, 42] due to its challenging nature and its importance in applications such as image search and retrieval, image annotation, video summarization, and human-computer interaction (HCI), to name a few. A comprehensive survey was performed recently in [43], where existing action recognition methods are categorized based on low-level features and high-level cues used for still image-based action recognition. Here, we present the high-level cues and the low-level features, accordingly.

2.1.1 Low-Level Features

The dense sampling of scale invariant feature transform (DSIFT), histogram of oriented gradient (HOG), shape context (SC) and GIST are the typical low-level features. A dense sampling of

the gray scale images is fulfilled to extract low-level features for action analysis, using the Scale Invariant Feature Transform (SIFT) method [44]. Proposed methods in [45, 32, 46, 47] used the DSIFT based feature to recognize action classes. The HOG feature proposed by [48] was used frequently for still image-based action recognition in [49, 50, 51, 52]. Shape context (SC) was proposed by [53] to extract shape features for object matching. The SC can help to detect and segment the human contour. Approaches such as [54, 51, 55] used the SC features for recognizing human actions. Extracting the SC feature is crucial for high-level cue representation of human body silhouettes for action recognition. GIST [56] method computes a set of holistic, spatial properties of the scene as an abstract representation of the scene . The GIST feature is mainly employed to integrate background or scene information. GIST has been used in works by [51, 57, 58] for action recognition in still images.

2.1.2 High-Level Cues

Human body, body parts, action-related objects, human object interaction, and the whole scene or context are considered as the most popular high-level cues for still image-based action recognition. Here, we present various high-level cues used for still image based action recognition.

2.1.2.1 Human Body

Human body is an important cue for still image-based action recognition. Human body can be detected manually [59] or automatically labeled [49, 42] in images. For instance, Li *et al*. [60, 58] manually selected and segmented minimum bounding box containing enough visual information for identifying the human body for action analysis in a still image. Delaitre *et al*. [32] extract features in areas within or surrounding the human bounding boxes. Contour [54] and poses [50] are other kind of information from the human body. In addition to body shape and bounding boxes,

the human body pose is also useful to extract the cue from body images. Thurau and Hlavac [50] and Ikizler *et al*. [61] used human body poses based on extracting a set of Non-negative Matrix Factorization (NMF) [62] bases. This can be challenging due to the limited number of poses they can detect and also the fact that many different human actions share almost the same poses. Semantic features such as the attributes [63, 31] can also be used to describe the actions in images with human body.

2.1.2.2 Body Parts

Body parts can be more related to action execution compared to the whole human body when performing different actions. Delaitre *et al*. [32] combined the results from a body part detector with other features based on the spatial pyramid bag-of-features. Poselet [64] is usually extracted from body parts, and can capture the salient body poses specific to certain actions. The studies in Maji *et al*. [65] and Zheng *et al*. [66] use the poselet to analyze human body parts for action recognition in still images. A graph model can simply represent the connections and the relations between different body parts. Yang *et al*. [67], Raja *et al*. [68] and Yao and Fei-Fei [69] considered a graphical model consisting of key-points in the human body as a set of nodes, as well as a set of edges that depict spatial relationships between the nodes.

2.1.2.3 Object

Some actions may involve the presence of different objects. Researchers have realized that knowing the related objects can help to recognize the corresponding actions. Objectness [70] is used in Prest *et al*. [57] to calculate the probability of a certain patch being an object. Sener *et al*. [71] proposed extracting several candidate object regions and using them in a Multiple Instance Learning (MIL) framework [72] .Yao *et al*. [31] used the Deformable Parts Model [73] to train an object

detector by using the ImageNet [74] dataset with provided bounding boxes. In [75], the input images are decomposed into groups of recognized objects. A language model is then used to describe all possible actions in the configurations of objects. While approaches in [55, 76, 77, 77] used object detectors to determine the occurrence of individual objects for action recognition some works integrate objects with the scene as context [66].

2.1.2.4 Human-Object Interaction

Rather than modeling the co-occurrence of humans and objects separately, the interaction between humans and objects is also useful for action recognition in still images, *i.e*. the relative position and angle between a person and the action related object. A mixture model of the relative spatial locations between the person's bounding box and the object's bounding box in still images is learned in [65]. In [55], a graphical modeling of the HumanObject Interaction (HOI) is presented by modeling the spatial relationship between the object and the human body parts as well as the dependence of the object with its corresponding image evidence. Rather than being limited to the interactions between one person and one object, the model in [55] was extended to deal with any number of objects [77, 78]. Approaches in [79, 31, 78, 57, 76] proposed to model human-object interaction by relying on the presence and detection of objects, as additional contextual information. This poses a challenge in cases where the action involves only a human, *i.e*. with no object interaction.

2.1.2.5 Context or Scene

While the background in an image is often taken as the context or scene of an executed action, the whole image also could be considered as the context or scene for action analysis, especially when the foreground (*e.g*. , human and object) occupy a relatively small area in the still image.

Some actions are mainly performed in certain scenes, *e.g*. diving in water, and driving on the road. Therefore, extracting information from the action context or the whole scene can be helpful for still image based action recognition [66]. In [32], they showed that the integration of spatial pyramids of background with human bounding boxes gives an improved performance. The occurrences of action-specific scene and objects using spatial information is introduced in [45] to recognize actions. Approaches in [51, 57, 56] extracted features from the whole image to encoded the scene for action image analysis. Although context or scene is useful for action recognition, noisy and cluttered background may have negative effects on action analysis. Furthermore, the context or scene may not provide helpful information for action recognition when different actions are performed in the same or similar scene.

2.1.3 Other Recent Methods

Recent methods [40, 41, 42, 80] on single image action recognition typically represent actions only by spatial features. For instance, in [81] a model using a collection of discriminative templates with associated scale-space locations is proposed. In [82] a hybrid algorithm using the Deformable Part Model is proposed to detect the human body parts or objects to recognize actions using the locations of the body parts and the objects. However, we advocate that the salient parts of the body along with their predicted motion can play crucial roles for action recognition.

Moreover, action classification in still images has recently benefited from CNN models [41, 83, 40, 20, 84], which offer an outstanding performance. The tradeoff, however, is that training CNNs requires learning millions of parameters and often a huge number of annotated images [85, 86, 87]. This poses a challenge when limited training data are available. CNNs are high-capacity classifiers with very large number of parameters that must be learned from training examples. Action recognition in still images suffers from lack of annotated images for a wide range of action

classes. Hence, recent works on single image action recognition focus on a limited number of action classes and primarily on human-object interactions. This poses a challenge when the action merely involves a human with no object interaction.

2.2 Ensemble learning

Reliable ensembles of classifiers and their method of construction have been an active research area in the context of supervised learning [88]. These methods involve constructing meta-algorithms, combining an array of machine learning techniques, resulting in one model in an attempt to decrease variance (bagging), bias (boosting), or improve the accuracy of models predictions (stacking). Ensemble methods are mainly attractive based on their premise that they offer higher accuracy when compared against individual classifiers making them up [88, 89].

In [90] a meta-level learning method is proposed using stacking for combining classifiers. In their proposed method, meta decision trees (MDTs) have base-level classifiers in the leaves instead of the class-value predictions. Properties of the probability distributions predicted by these base-level classifiers (such as entropy and maximum probability) are then considered as the meta-level attributes, rather than the distributions themselves. The confidence of the base-level classifiers are reflected by the properties of the probability distributions predicted by the base-level classifiers, such as entropy and maximum probability. These properties result in considerably small MDTs, which can be inspected and further interpreted. In [91], a well-known stacking method called SCANN is proposed to leverage correspondence analysis in order to detect possible correlations among the predictions of the base classifiers. The The class-value predictions as the original meta-level feature space is further transformed in order to remove these correlations. A nearest neighbor method is employed as the meta-level classifier on the new feature space.

It is naturally expected that ensembles of classifiers generated by stacking perform superior results than the best individual base-level classifier. The additional effort in constructing such complicated models would otherwise have no valid justification. In chapter 7, we demonstrate that constructing ensembles of two heterogeneous learning algorithms, a deep learning classifier and a zero-shot classifier, improves the action classification accuracy by a considerable margin.

2.3 Still Image Datasets for Action Recognition

Most popular action classification datasets, such as KTH [92], Weizmann [93], Hollywood-2 [94], HMDB [95], UCF101 [96] consist of short clips, manually trimmed to capture a single action in videos. They serve a valuable purpose, but address a different need than what UCF-STAR has to offer. Moreover, most of the large image datasets such as Caltech [97], Pascal VOC [59], and ImageNet [74] have been created for the purpose of object classification tasks, but not for action classification.

Action images in sports [51, 54, 45] are the earliest and of the most popular usage for recognition probably due to the relatively small human pose variations within the same actions in sports activity, and the distinctiveness and uniqueness of specific sports actions in single images. Daily activity datasets [65, 31, 32, 75] contain common activities performed by humans in daily life. Pascal VOC [65] competition includes still image-based action recognition starting from 2010. The nine actions include phoning, playing an instrument, reading, riding a bicycle or motor-cycle, riding a horse, running, taking a photograph, using a computer, or walking. Later in 2011, one more action called jumping was added to the original 2010 dataset. Additionally, there are people labeled with the action 'other', meaning none of the above actions are being performed. There is a minimum of around 200 people per action category. Actions are not mutually exclusive, for example a person may simultaneously be walking and phoning. The 'other' action is mutually

exclusive to all other actions. In 2012, the dataset was expanded again: about 90% increase in size over VOC 2011. There is a minimum of around 400 people per action category. UCF-*STAR* differs from them, as we provide human body-motion action categories rather than relatively stationary actions such as reading, phoning, taking a photograph and using computer.

Datasets by [32], [61], [98], [60], [79], [47] contain 968, 467, 2,458, 2,400, 341 and 2,100 images, respectively. Images were collected from different sources like Google Image Search, Flickr and PASCAL VOC 2010 to build three to seven action categories. The main differences with UCF-*STAR* dataset are the small number of action classes and the small number of overall images. Furthermore, classes contain actions with less human body motion *i.e*. playing/holding instruments and wearing hat; which are not the primary focus in UCF-*STAR*.

Thurau and Hlavac [50] and Raja *et al*. [68] used still images extracted from the popular action videos [93, 92] to build ten and six action classes, respectively. The extracted image frames usually have a relative static or cleaner background.

Yao *et al*. [31] collected a challenging and relatively larger dataset, called Stanford 40 Actions, containing 40 diverse daily human actions. All images were obtained from Google, Bing, and Flickr. There are 9,352 images in total. Le *et al*. [75] assembled a dataset from 11,500 images of the PASCAL 2012 VOC trainval set, selecting all those images representing a human action, resulting in 2,038 images over 89 action classes. However, the number of images in each class is extremely small.

The demanding nature of deep learning applications to train a robust predictive model require a large number of images for each class. This motivated us to construct the *UCF-STAR* dataset containing 1,038,622 images spread over 50 different action classes, with a large number of images per class. It is also worth mentioning that *UCF-STAR* provides not only action labels for each image, but also includes a rich set of metadata further explained in the section 6.

CHAPTER 3: STILL IMAGE ACTION RECOGNITION BY PREDICTING SPATIAL-TEMPORAL PIXEL EVOLUTION

In this chapter [1], we introduce a new latent space-time domain mapping technique as well as a deep learning method for action recognition in still images based on their latent space-time representations.

3.1 Mapping to the Latent Space-Time Domain

For identifying an action not all pixels carry the same importance. Some pixels capture less meaningful information or even carry misleading information, while others carry more discriminative information. Here we describe the domain mapping stage that projects data from the source domain of static RGB images onto the latent space-time domain $Rank_{\text{SM-POF}}$, by ranking pixels based on their spatial and predicted temporal significance. This leads to a significant reduction in noise by focusing the classifier's attention to important pixels.

The *Domain Mapping* stage itself consists of two steps that are depicted in Figure 3.1.1. *First*, we construct a 3-way tensor, $\mathcal{Q} \in \mathbb{R}^{P \times F \times 3}$, for each action class $a_n \in A$, where $A = \{a_1, ..., a_N\}$ is a discrete set of N actions. $\mathcal{Q}$ is a three-channel feature map consisting of the spatial saliency map and the predicted horizontal and vertical optical flow components for all images in an action class $a_n \in A$. P and F represents the number of pixels in each image and the number of images in each action class a_n, respectively. *Second*, tensor $\mathcal{Q}$ is projected onto a new domain $Rank_{\text{SM-POF}}$, providing a compact representation of the spatiotemporal attributes using an ordinal regression to

[1] This content is reproduced from the following article: Safaei, Marjaneh, and Hassan Foroosh. "Still Image Action Recognition by Predicting Spatial-Temporal Pixel Evolution." In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 111-120. IEEE, 2019.

show the extent of contribution of a pixel for recognizing an action. Below is the description of these two steps and the motivation/intuition behind each.

3.1.1 Forming Tensor $\mathcal{Q}$

$\mathcal{Q} \in \mathbb{R}^{P \times F \times 3}$ is a feature space tensor consisting of spatial and temporal information related to the underlying action for all training images in the action class a_n. Each column in tensor $\mathcal{Q}$ corresponds to an image I. In tensor $\mathcal{Q}$, the first channel represents the Sailency Map (*SM*) of images, the second and the third channel represent the Predicted Horizontal Optical Flow (POF_{h}) and the Predicted Vertical Optical Flow(POF_{v}) for each pixel in images, respectively.

For the temporal information, we use a CNN model similar to the one proposed by [99] to predict a dense optical flow. This optical flow map represents how and where each pixel in the input static image is predicted to move. For this purpose, the optical flow vectors are first quantized into 40 clusters by *k*-means. The problem is then treated in a manner similar to semantic segmentation, where each region in the image is classified as a particular cluster of the optical flow. These clusters are then used to predict the motion direction for each pixel. A softmax loss layer at the output is then used for computing gradients. We generate the output as softmax probabilities over the optical flow vectors for each pixel. The softmax loss is spatial, summing over all the individual region losses. This leads to an $M \times N \times C$ softmax layer, where M, N and C represent the number of rows, columns and clusters, respectively. Let I represent the image and Y be the ground truth optical flow labels represented as quantized clusters. Then the spatial loss function $L(I, Y)$ as defined by Walker *et al*. [99] is given by:

$$L(I, Y) = -\sum_{i=1}^{M \times N} \sum_{r=1}^{C} \mathbb{1}(Y_i = r) \log F_{i,r}(I), \tag{3.1}$$

where $F_{\mathrm{i,r}}(I)$ represents the probability that the i_{th} pixel will move according to cluster *r*, and $\mathbb{1}(Y_{\mathrm{i}} = r)$ is an indicator function. A problem with this loss function is that it implicitly assumes a uniform probability mass function (pmf) for the motion clusters, which is very unlikely and prone to noise. Therefore, we modified Eq.3.1 in order to minimize the noise by taking into account only the k most-likely clusters, *i.e*. the k clusters with the highest probability and optimized the pre-trained CNN [99] using our custom loss function in Eq.3.2. This amounts to replacing the second summation in Eq.3.1 with an order statistic filter as follows:

$$\hat{L}(I, Y) = -\sum_{i=1}^{M \times N} \sum_{r=1}^{C} \omega_r P_{i,(r)}, \tag{3.2}$$

where ω_r are some weight factors, and

$$P_{i,(r)} = \mathbb{1}(Y_i = (r)) \log F_{i,(r)}(I) \tag{3.3}$$

is the pmf in descending order of values, *i.e*. $P_{i,(1)} \geq P_{i,(2)} \geq ... \geq P_{i,(C)}$. We set $k = 10$ and assume $\omega_r = \frac{1}{k}$ for $P_{i,(1)}, \ldots, P_{i,(k)}$, and $\omega_r = 0$, otherwise. This is equivalent to averaging over the probabilities of the k most-likely clusters. The two components of the predicted optical flow in this manner are then used as the second and third channels in tensor $\mathcal{Q}$, *i.e*. POF_{h} and POF_{v}.

The first channel of $\mathcal{Q}$, represents the static saliency map of the image using a bottom-up approach [100], where each pixel indicates the statistical likelihood of saliency of a feature matrix given its surrounding feature matrices. We set all the values below some threshold τ in the *SM* channel to zero, in order to ensure a better localization and representation of the shape of the salient regions of the image. The threshold τ was selected automatically using Otsu's method [101].

The *SM*, POF_{h} and POF_{v} for training images in action class a_n, are then normalized in the unit interval and concatenated to form tensor $\mathcal{Q}$ for each action a_n. We denote each column in tensor

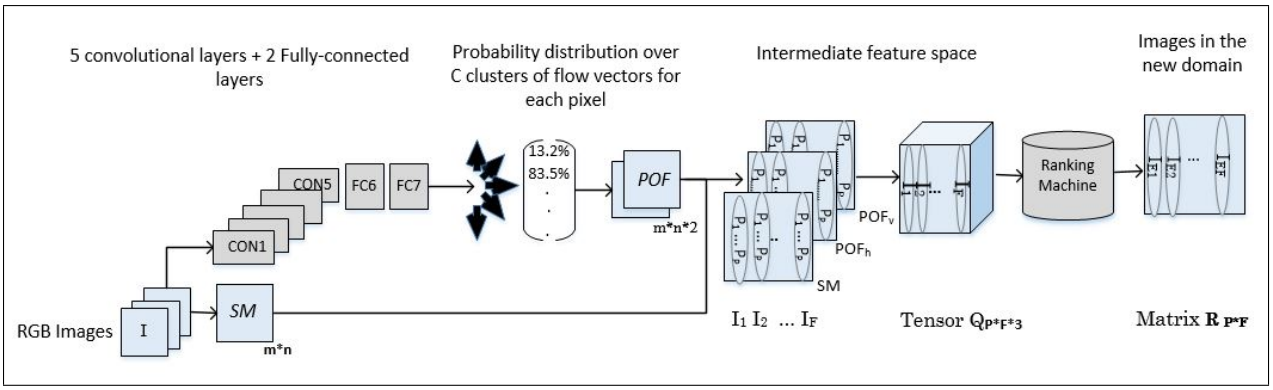

Figure 3.1: Domain Mapping process. Step 1: Converting RGB images to the intermediate feature space tensor $\mathcal{Q}$ by concatenating their saliency map (*SM*) and predicted optical flows (*POF*). Step 2: Mapping tensor $\mathcal{Q}$ onto matrix $\mathcal{R} \in Rank_{\text{SM-POF}}$, presenting new image representations through *Rank-SVM* algorithm

$\mathcal{Q}$ as c such that $\forall c_{\text{i,j}} | i \in F, j \in [1,2,3]$. For instance, $c_{\text{i,1}}, c_{\text{i,2}}$ and $c_{\text{i,3}}$ represent the *SM*, $\textit{POF}_{\text{h}}$ and $\textit{POF}_{\text{v}}$ features of the $I_{\text{i-th}}$ image, respectively.

Tensor $\mathcal{Q}$ may be viewed as a noisy or approximate prediction of the spatiotemporal attributes of the action. The inaccuracies are of course due to the nature of the prediction process of spatiotemporal attributes in a single image, which is a lot less accurate than estimating them when video data is available. Therefore, our next goal is to regress a more compact and distinctive representation of the spatiotemporal information in images, described in the next section.

3.1.2 Mapping to $Rank_{\textit{SM-POF}}$ Domain

As described in section 3.1.1, tensor $\mathcal{Q}$ represents still images in action class a_n, by concatenating their SM and POFs components, which are inherently error prone. The key to the success of this task is how to extract discriminative spatiotemporal features to efficiently model the pixels evolution. We further borrow inspiration from [102] to learn pixels evolution from spatial to tem-

poral, which we show it is an important cue for classifying actions. This is, essentially, a ranking process, where the projection kernel is learned from a training set. The parameters of the linear ranking functions encode the pixel evolution in a principled way. To learn such ranking machines, we use the supervised learning to rank [103]. We propose to use the parameters of the ranking machine as the new spatiotemporal image representation to characterize the extent of significance of spatiotemporal attributes of each pixel in recognizing an action.

Let $\mathbf{V} = [v_{t_1}, v_{t_2}, v_{t_3}]^T$ represents column j in the 3-channel tensor $\mathcal{Q}$, such that $v_{t_1} = c_{j,1}, v_{t_2} = c_{j,2}$ and $v_{t_3} = c_{j,3}$. Vector V denotes the feature vector, whose rows are the j_{th} columns in tensor $\mathcal{Q}$, representing *SM*, POF_{h} and POF_{v} for the j_{th} image in action class a_n.

A linear Rank-SVM is basically a pairwise linear ranking machine that learns a linear function or mapping of the form $\Psi(\mathbf{V}; \mathbf{E}) = \mathbf{E}^T\mathbf{V}$ [103, 104]. We assume that the order of the sequence in the training set V is: $v_{t_1} < v_{t_2} < v_{t_3}$ *i.e. SM* is always the first channel, followed by the second channel POF_{h} , and finally the third channel POF_{v} in tensor $\mathcal{Q}$. The problem of learning the optimal linear kernel for V reduces to solving the following convex optimization problem [103]

$$\arg\min_{\mathbf{E}} \frac{1}{2}\| \mathbf{E} \|^2 + W \sum_{\forall v_{t_i}, v_{t_j}, v_{t_i} \geq v_{t_j}} \epsilon_{ij} \tag{3.4}$$

$$s.t. \quad \mathbf{E}^T(v_{t_i} - v_{t_j}) \geq 1 - \epsilon_{ij}, \quad \epsilon_{ij} \geq 0, \tag{3.5}$$

where ϵ_{ij} are slack variables and W is a regularization parameter. By solving the above optimization problem, we learn a vector of parameters E, which encodes the order constraints. In our case, E may be interpreted as representing the evolution of pixels from appearance to latent motion in the still image. Upon reshaping E to the original image size, we obtain the latent space-time image, denoted as I_E. This latent image representation is used to construct the *STCNN* classifier

which will be introduced in the next section. Our experiments later demonstrate that the proposed latent representation, $Rank_{\text{SM-POF}}$, has superior discriminative properties for action recognition in still images, compared with the raw image data, especially for actions involving substantial body motions.

3.2 STCNN Classifier

What sets our work apart from existing efforts, in this context, is that we propose to use not only spatial cues but also predict temporal patterns for recognizing human actions in still images. We define *spatial* as the salient human body pixels that describe the properties of human actions, while the term *temporal* represents the predicted optical flow for pixels in still images. In our method the characterization of the underlying human action does not require any bounding box annotations. In contrast, approaches adopted by related works require additional input, *e.g*. Delaitre *et al*. [32] extract features in areas within or surrounding the human bounding boxes. The Space-Time CNN (*STCNN*), is constructed through transfer learning of action classes by fine-tuning an appearance-based (spatial only) CNN model on the proposed latent space-time domain $Rank_{\text{SM-POF}}$ presented in section 3.1. In section 1.1 we describe the *STCNN* in detail.

Here we propose a deep CNN that is trained to classify human actions based on both spatial and predicted temporal features, *i.e*. using the images in the $Rank_{SM-POF}$ as input. We take a network trained on a different domain for a different source task, and adapt it for the proposed domain $Rank_{SM-POF}$ and the new target task, which is action classification in single images. We do a supervised domain adaptation via fine-tuning the pre-trained network on the new domain.

We also created a new large dataset of *2M* still images, UCFSI-101, by sampling random video frames from the UCF-101 dataset [96]. Our experiments on UCFSI-101, Willow [32], Stanford-

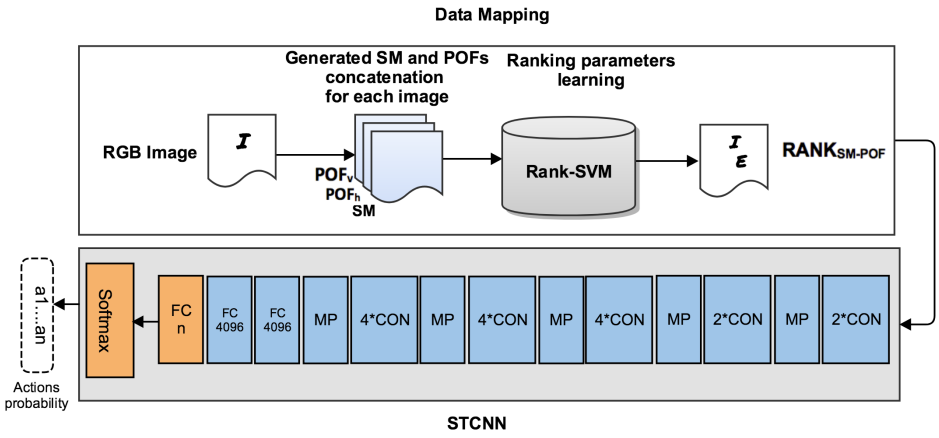

Figure 3.2: Schematic overview of the proposed framework. Domain Mapping: RGB images are mapped to the new proposed domain $Rank_{SM-POF}$. CNN action classifier: Deep features from the activation of the fully connected layers of a STCNN model are used as input to a softmax layer, modified to fit our domain, to determine the final classification.

40 [31], WIDER [105], BU_{101} [29] datasets, as well as our newly collected still images from the wild demonstrate that the proposed domain mapping method is extremely effective in capturing action attributes in still images. As a result, *STCNN* outperforms the state-of-the-art methods by a significant margin. Fig. 3.2 depicts the architecture of the proposed approach.

3.3 STCNN Architecture

We build our model on top of a CNN, which learns to selectively focus on spatial-temporal features and the pixels evolution from spatial to temporal space. Rather than having two independent spatial-CNN and motion-CNN to capture appearance and motion features separately [106], our proposed method presents a joint unified spatial-temporal CNN, named *STCNN* trained on $Rank_{SM-POF}$ to capture the spatial-temporal combined features and classify actions accordingly. Our STCNN is similar to the architecture proposed in [107]. This network is formed from sixteen

successive convolutional layers followed by three fully connected layers. We denote the convolutional layers as CON(*k*,*s*), indicating that there are *k* kernels, of size $s \times s$. We also denote the max-pooling layer as MP. The input to our STCNN is a fixed-size 224×224 image. The convolution stride is fixed to 1 pixel. Max-pooling is performed over a 2×2 pixel window, with stride 2. Finally, FC(*n*) denotes a fully connected layer with *n* neurons. Our network architecture can be described as: 2*CON(64,3) $\rightarrow$ MP $\rightarrow$ 2*CON(128,3) $\rightarrow$ MP $\rightarrow$ 4*CON(256,3) $\rightarrow$ MP $\rightarrow$ 4*CON(512,3) $\rightarrow$ MP $\rightarrow$ 4*CON(512,3) $\rightarrow$ MP $\rightarrow$ FC(4096) $\rightarrow$ FC(4096) $\rightarrow$ FC(n).

The model in [107] was trained on Imagenet [74]. In order to accomplish transfer-learning by adapting it to the $Rank_{SM-POF}$, we changed the *FC8* layer in order to adapt it to our target domain classes. In all our experiments, we kept some of the earlier layers fixed, and fine-tuned some higher-level portions of the network, and finally trained the new classifier layer *FC8* on the target dataset from scratch. Even though it appears that we can afford to train a network from scratch when the target dataset is large enough, in practice it is quite often still beneficial to initialize the weights from a pre-trained model. In this case, we have enough data and confidence to fine-tune through the entire network. We use a smaller learning rate for layer weights that are being fine-tuned, in comparison to the weights for the new linear classifier that computes the class scores. Our goal was to learn a mapping between the $Rank_{SM-POF}$ and the action class.

We benefit from CNN's outstanding classification capability, through supervised domain adaptation by fine-tuning a network specifically pre-trained for appearance based classification on RGB images, to train the STCNN for appearance-motion (action) based classification on $Rank_{SM-POF}$ domain. We further evaluate the proposed *STCNN* in the next section.

3.4 Experiments

In this section, we demonstrate the effectiveness of our proposed domain-mapping approach along with the proposed *STCNN* network for single image action recognition. We ran experiments on UCFSI-101, Willow, Stanford-40, WIDER, BU_{101}, and finally our newly created dataset from the *wild*, details of which are described below. We report the results in terms of average precision (AP), where for each class we compute the average precision over all action classes. For all previously existing datasets, we used the train and test splits provided by the original authors. Our comprehensive experimental evaluations and results are elaborated in the following sections.

3.4.1 Datasets

UCFSI-101. We created a large annotated ***still-image*** dataset , by extracting over *2M* frames randomly from the original UCF-101 video dataset [96]. UCF-101 has 13,320 videos from 101 action categories that spread across 5 broad groups, that is (1) Human-Object Interaction, (2) Body-Motion, (3) Human-Human interaction, (4) Playing Instruments and (5) Sports. Not all frames are useful for image-based action recognition. Therefore, after the sampling process, we eliminated frames with no human subjects clearly visible. We collected 1,585,071 frames to serve as our training set and labeled them based on the video labels they belong to, as well as 617,321 frames to help form our test set. Our train/test frames are not extracted from the same videos. In *UCFSI-101*, visual variance of extracted frames from the same video depends on action categories. Action classes with fast body motion and different key action phases, *e.g*. Diving, have high visual variance compared to actions that are relatively stationary.

While most related works perform their experiments on datasets with low number of classes, focused on human-object interactions, our method mainly focuses on the human body motion with

100 classes. Considering that our method is based on the patterns associated with the human motion and also the overall shape and location of the most salient parts in the human body, we naturally expect to get better results on Body-Motion categories as opposed to other categories that are highly dependent on detecting the presence of a specific object in the scene.

Willow [32] action dataset contains 911 images split into seven action categories: Interacting with computer, Photographing, Playing music, Riding bike, Riding horse, Running and Walking. We used the train and test splits provided by the original authors. We also used standard data augmentation, *i.e*. randomly mirrored images to avoid spatial biases (such as humans always centered in the image). We further divided the 7 action categories into two main groups, Body-Motion and Non-Body-Motion actions. The first three actions, pointed out in the beginning of this paragraph are considered as Non-Body-Motion actions since they are highly dependent on human-object interactions, and the rest are considered as Body-Motion actions.

Stanford-40 dataset [31] is a large and challenging dataset. It consists of 40 actions and 9,532 images. In each category, 100 images are used for training and the others are used for test. We divided this dataset to 2 categories: (1) Body-motion and (2) Non-body motion, with 11 and 29 actions respectively. Climbing, Jumping, Cleaning-floor, Riding-bike, Riding-horse, Rowing-boat, Running, Walking-dog, Shooting-arrow, Throwing-frisby and Waving-hands are considered as Body-motion and the rest as Non-body motion human action classes.

WIDER [33] attribute dataset includes 14 human attribute labels and 30 event class labels containing 13,789 images with 57,524 person bounding boxes. We then considered 6 actions under the Body Motion category; *i.e*. running, basketball, football, soccer, skiing and hockey.

BU$_{101}$ [29] consists of $23.8K$ action images that correspond to the 101 action classes in the UCF101 video dataset. Action categories are divided into five types: Human-Object Interaction, Body-Motion, Human-Human Interaction, Playing Musical Instruments, Sports. For each action

class, images are downloaded from the Web using corresponding key phrases, and then manually remove irrelevant images or drawings and cartoons. We used the train and test splits provided by the original authors for all datasets.

UCF-STAR [2]. We introduce this new dataset, which is the largest annotated still image dataset in the literature in chapter 6. Images in *UCF-STAR* are equipped with rich and structured metadata making our dataset particularly suitable for research on multi-modal applications, e.g. in multi-media. All classes in *UCF-STAR* are of sufficient size to avoid issues with training models. Our resulting benchmark consists of a total of 664,718 training images, 166,180 validation images, and 207,724 test images from 50 classes. *UCF-STAR* construction pipeline and its statistic are extensively discussed in chapter 6.

3.4.2 Evaluating the Role of Domain-Mapping

In order to evaluate the significance of our proposed latent representation domain, $Rank_{\text{SM-POF}}$, we performed substantial experiments on two datasets, by comparing the action classification accuracy when we consider 1) only spatial features 2) only predicted temporal features 3) the intermediate spatiotemporal tensor $\mathcal{Q}$, and 4) the proposed latent spatiotemporal representation in the $Rank_{\text{SM-POF}}$ domain.

First, we trained STCNN over RGB images on both UCFSI-101 and *UCF-STAR* training sets from scratch. As expected, action classification using only raw RGB images proves to be a very challenging task due to the overwhelming level of variability in terms of various body poses that represent the same action, appearance, viewpoints, background, etc.

Second, we converted RGB images into their Saliency Maps (*SM*) in order to only focus on the

[2] *UCF-STAR* dataset will be fully discussed in chapter 6

salient pixels. While focusing only on the human salient pixels reduces the recognition errors slightly, *SM* alone did not prove to be a good representation either for still image action recognition. So far, we have relied only on spatial features, *i.e*. raw RGB or salient pixels, knowing that lack of temporal information is a major gap in still images action recognition.

Third, we generated the Predicted Optical Flow (*POF*) of each pixel as discussed in section 3.1.1 in order to learn an action classification model based on temporal features only. As expected, training the proposed STCNN only on *POF* data did not lead to a noticeable performance gain either.

Fourth, with the assumption that for still images both spatial and latent temporal information are crucial for recognizing human actions, we trained STCNN on the tensor $\mathcal{Q}$, which is an intermediate spatiotemporal representation of action images, as discussed in detail in section 3.1.1. Experiments show that incorporating both spatial and latent temporal information through $\mathcal{Q}$ improves the results by a considerable margin. Intuitively, using saliency map helps detect the most salient regions in the image, while the predicted optical flow helps identify the most probable motions for those salient pixels.

Finally, STCNN was trained on the proposed latent space-time representation I_{E}, as explained in section 3.1.2. Extensive experiments on UCFSI-101 and *UCF-STAR* datasets show that this representation substantially outperforms all other representations. This is due to the fact that using Rank-SVM we specifically learn a feature space that captures both the spatial relations between salient pixels and their corresponding latent temporal evolution. Although this information might be present in the raw space-time tensor $\mathcal{Q}$, it is convoluted with irrelevant information and noise. As a result the learned representation space exhibits higher discriminative properties since the features are specifically learned from raw data for the purpose of still image action recognition. Fig. 3.3 summarizes all above experiments side-by-side for comparison.

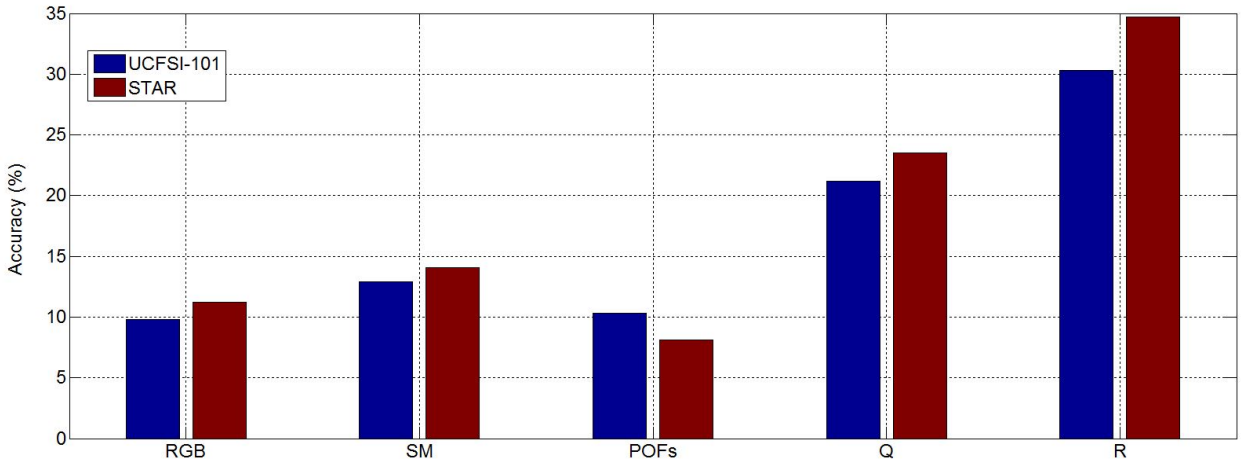

Figure 3.3: Action classification accuracies by training STCNN from scratch on different domains. Mapping to the proposed latent spatiotemporal domain, $\mathcal{R} \in Rank_{\text{SM-POF}}$, improves the accuracy by a big margin.

3.4.3 Evaluation of Transfer Learning

The advantage of the proposed domain-mapping approach was fully evaluated in section 3.4.2. After mapping the still images from RGB onto I_{E} we then train our deep STCNN network on training sets associated with the target domain. All our experiments are implemented in caffe [108] and are run on a single GeForce GTX Titan GPU with 15GB of memory.

We expect that CNNs learn more generic features on the bottom layers of the network, and more convoluted dataset-specific features near the top layers of the network. Therefore, we considered two different approaches for our transfer learning experiments by fine-tuning a pre-trained CNN for appearance classification [107] to perform action classification . (1) *Fine-tuning all layers.* Under this approach, we re-trained all network parameters, including all convolutional layers on the bottom of the network; (2) *Fine-tuning the last seven layers.* Rather than only re-training the final classifier layer from scratch, we performed fine-tuning on the last seven layers. In all

experiments, as shown in tables 3.1 and 3.2, fine-tuning the last 7 layers of the network increased the performance on all action categories. Also among all groups, we obtained the best results on the body-motion group, in which actions heavily depend on the predicted motion and also the most salient part of the body. As presented in tables 3.1 and 3.2, transfer learning method also yields more promising results compared to the case where a CNN is trained from scratch on $Rank_{\text{SM-POF}}$ domain.

Table 3.1: Transfer learning evaluation on UCFSI-101.

Model	Train from scratch		Fine-tune all layers		Fine-tune 7 layers	
	Top-1 accuaracy	Top-5 accuaracy	Top-1 accuaracy	Top-5 accuaracy	Top-1 accuaracy	Top-5 accuaracy
RGB	9.8%	11.1%	10.3%	16.8%	13.5%	17.9%
SM	12.9%	19.8%	17.2%	26.8%	20.5%	29.8%
POFs	10.3%	15.7%	12.9%	13.8%	14.6%	20.1%
$\mathcal{Q}$	21.2%	35.0%	41.8%	55.3%	64.6%	71.8%
$I_{\text{E}} \in \mathcal{R}$	30.3%	38.9%	46.6%	58.7%	**69.5%**	**79.9%**

Table 3.2: Transfer learning evaluation on *UCF-STAR*.

Model	Train from scratch		Fine-tune all layers		Fine-tune 7 layers	
	Top-1 accuaracy	Top-5 accuaracy	Top-1 accuaracy	Top-5 accuaracy	Top-1 accuaracy	Top-5 accuaracy
RGB	11.2%	15.2%	12.3%	18.8%	15.5%	20.9%
SM	14.1%	20.8%	16.2%	25.6%	21.7%	30.6%
POFs	8.1%	11.7%	10.5%	19.2%	16.6%	20.9%
$\mathcal{Q}$	23.5%	34.0%	45.7%	58.3%	67.0%	75.6%
$I_{\text{E}} \in \mathcal{R}$	34.7%	39.9%	47.6%	60.3%	**79.1%**	**85.8%**

3.4.4 Evaluation of STCNN Classifier

In this section we fully evaluate STCNN, a deep network fine-tuned over images in the $Rank_{\text{SM-POF}}$ latent domain, to perform action classification and compare it's performance against recent state-of-the-arts approaches. First, we compare STCNN with the commonly used late fusion strategy in [106]. Here, we implemented two spatial (*SM*) and temporal (*POF*) streams using deep CNNs separately. Next, we combined the softmax scores of each stream via late fusion in order to fully validate the merits of STCNN, a joint spatiotemporal network. Table 3.3 represents superior

Table 3.3: STCNN comparison against the two-stream late fusion approach.

Dataset	**Two-stream [106]**		**STCNN**	
	Top-1 accuaracy	Top-5 accuaracy	Top-1 accuaracy	Top-5 accuaracy
UCFSI-101	19.7%	24.3%	69.5%	79.9%
UCF-STAR	25.6%	39.9%	79.1%	85.8%

STCNN results against the two-stream late fusion[106] strategy on two different datasets. Therefore, a unified spatiotemporal representation resulted as spatiotemporal pixel's evolution improve the classification performance by a considerable margin.

Table 3.4: Results (mAP) on UCFSI-101 broken down by category groups.

	Intermediate feature space tensor Q			**After domain mapping Rank$_{\text{SM-POF}}$**		
STCNN	From Scratch	Fine-Tune all layers	Fine-Tune 7 layers	From Scratch	Fine-Tune all layers	Fine-Tune 7 layers
Human-Object	17.10	35.0	56.0	23.02	36.10	57.0
Body-Motion	42.0	**58.25**	**84.20**	49.0	**60.10**	**89.0**
Human-Human	35.0	42.04	58.03	42.02	44.0	60.0
Playing-Instrument	18.21	30.05	60.0	21.0	35.20	65.04
Sport	23.20	42.0	65.0	30.11	50.23	69.12

As discussed earlier I_E representations were generated for images in all datasets. We then removed the last fully-connected layer from the network, which was pre-trained on I_E representations of images in UCFSI-101 dataset, mentioned in section 3.4.3, and trained a linear softmax classifier for all other datasets. We stress that test and training images in all datasets including are mutually exclusive. Table 3.4 presents the Results on UCFSI-101 broken down by category groups.

As shown in tables 3.5-3.8, STCNN achieved promising results compared to state-of-the art approaches on all datasets. While a classifier solely based on appearance can be confused by actions appearing in similar contexts, the inferred motion provide cues about the fine-grained differences

among actions to aid recognition. Considering that we focus on the human body salient pixels and their motion direction, we achieved the best performance on the body-motion group on all datasets.

Table 3.5: Results (mAP) on Willow dataset.

Method	**Non-Body-Motion**	**Body-Motion**
Delaitre *et al*. [32]	55.6	62.7
Delaitre *et al*. [76]	55.4	70.7
Sharma *et al*. [109]	59.0	71.1
Sharma *et al*. [81]	60.1	73.2
Khan *et al*. [110]	62.4	72.2
Khan *et al*. [111]	64.1	78.05
Zhao *et al*. [80]	67.8	79.3
Khan *et al*. [42]	62.2	76.0
STCNN	64.5	**78.7**

Table 3.6: Results (mAP) on Stanford-40.

Method	**Body-Motion**	**Non-Body-Motion**	**All**
Gkioxari *et al*. [112]	93.87	89.73	90.9
Khan *et al*. [111]	56.92	51.51	53
Khan *et al*. [42]	53.51	51.28	51.9
Yan *et al*. [113]	92.26	87.07	88.5
Zhao *et al*. [114]	-	-	83.4
Zhao *et al*. [80]	-	-	54.5
Zhao *et al*. [115]	-	-	80.6
Zhou *et al*. [116]	-	-	55.3
Sharma *et al*. [117]	-	-	72.3
Khan *et al*. [118]	-	-	75.4
Gao *et al*. [119]	-	-	74.9
STCNN	**94.3**	73.1	81.76

While [111] constructs the spatial pyramids on the full-body, upper-body and face bounding boxes, temporal information is completely ignored in their method. As shown in table 4.3, [111] performs poor on Stanford-40 compared to Willow dataset. The large number of action categories in Stanford-40 makes this dataset particularly challenging. There are many common human body

Table 3.7: Results (mAP) on WIDER.

Method	**mAP**(%)	
RCNN	80.0	
R*CNN	80.5	
DHC	81.3	
ResNet-SRN	86.2	
VeSPA	82.4	
STCNN	Body-Motion	Non-Body-Motion
	86.8	59.5

Table 3.8: Results (mAP) on BU_{101} dataset.

mAP (%)					
Category	Human-Object	Body-Motion	Human-Human	Playing-Instrument	Sport
STCNN	61.1	84.4	58.7	71.3	74.8

poses among different action classes which makes action recognition harder when temporal cues are ignored. As shown in tables 3.5 and 3.6, STCNN maintains its performance on both datasets.

Presence of particular objects and scene types [76, 32] as well as their spatial and scale relations with humans can characterize the action in the images, however, many human action images with no specific object involved and no distinguishable scene, pose a serious challenge for current human-object interaction methods. The proposed methods in [109, 82, 81] have mainly relied only on body parts and poses as cues for action recognition. The extensive manual labeling of human bounding boxes and having enough number of body pose and parts detectors, as well as the fact that different actions might have very similar poses can be very challenging. As shown in all experiments focusing on pixels' evolution within a generated spatiotemporal image representation improves action classification in still images by a significant margin, especially on body-motion group.

We carried on a comprehensive analysis on all components of our proposed architecture. Results on multiple benchmarks demonstrate that appearance and motion are complementary sources of information, hence using both leads to significant performance improvements in single image action recognition.

CHAPTER 4: A ZERO-SHOT ARCHITECTURE FOR ACTION RECOGNITION IN STILL IMAGES

In this chapter [1], we propose *ZTD*, a Zero-shot method to classify human actions through Tensor Decomposition, that to the best of our knowledge, has been only used for action recognition in video but not in still images. *ZDT* does not rely on *training*. Instead, it relies on selecting the maximum of a sequence of joint probability distributions in the proposed latent space-time domain, *i.e*. a maximum *a posteriori* estimate of the unknown class label. In ZTD, we implicitly model the latent space-time domain of each action class, using a group of images per action. The assumption is that, although a single image may not uniquely characterize its latent space-time domain, a group of sample images for an action would form a prototype tensor that would span the latent domain of the corresponding action class. Action classification for a test image is then treated as recognizing the prototype group whose low-rank representation is closest to the test image, *i.e*. by measuring to what extent the test image would perturb such low-rank representation for each class. *ZTD* involves the following three steps: (i) Forming action prototypes, (ii) Rank-1 subspace representation, (iii) Action classification. These steps are described in the following sections. Fig. 4.1 depicts the *ZTD* architect.

4.1 Forming Action Prototypes

As described in section 3.1, we generate a tensor $\mathcal{Q}$ for each action class $a_n \in A$ and use Rank-SVM to map $\mathcal{Q}$ onto a matrix $\mathcal{R}$. Each column of R then reshaped as an image I_E represents the pixelwise latent space-time features for the corresponding image in $\mathcal{Q}$.

[1] This content is reproduced from the following article: Safaei, Marjaneh, and Hassan Foroosh. "A zero-shot architecture for action recognition in still images." In 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 460-464. IEEE, 2018.

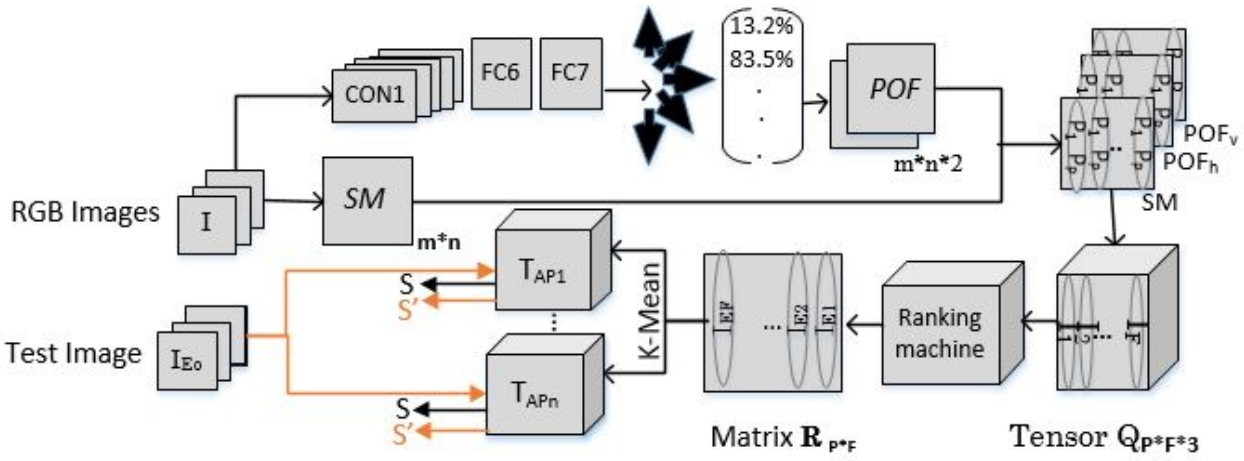

Figure 4.1: Converting RGB images to the intermediate feature space tensor Q by concatenating their saliency map and predicted optical flows (POF). Mapping tensor Q to matrix R, presenting new image representations through Rank-SVM. Action prototype tensors (T_{AP}) are formed and decomposed to generate the original signature vector S. $\acute{S}$ is generated after the I_{Eo} insertion.

Different actions have different number of key phases. For instance, Lifting has three key phases, beginning with the weight separated from the floor, bending arms at the elbow pulling under the bar, and finally lifting the bar over head with the arms completely straight. A Rank-SVM trained on images depicting the same action phase can be expected to lead to similar ranking functions, *i.e.* similar I_E representations. Therefore, by clustering the columns of the matrix $\mathcal{R}$ for each action a_n, we construct k action prototypes for that action class, where the columns grouped together in each cluster are estimated to belong to the same key-phase of that action.

We employ *k*-means for clustering the columns of R. One issue with *k*-means is the choice of k. For this purpose, we assume each action has a limited number of key-phases as mentioned before. Therefore, we set $k \in \{2, 3, 4\}$. We initialize the centroids so that they are as far apart as possible. We then use the *elbow* method [120] to determine the optimum number of clusters for *k*-means clustering. The basic idea behind the elbow method is that the optimal choice of k is the one that

minimizes the total intra-class variance.

Once the matrix $\mathcal{R}$ is divided into k clusters, the columns in each cluster reshaped as 2-D images of size $X \times Y$ (*i.e.* same size as the original images) yield the images I_E. For each cluster in each action, these images are stacked together to form k 3-way tensors per action. Each such tensor $\mathcal{T}_{ka_n} \in \mathbb{R}^{X \times Y \times Z}$ is referred to as the action prototype tensor for the k-th cluster of action class a_n, where Z is the depth of the tensor, *i.e.* the number of elements clustered together to form the prototype of a given phase in an action.

To summarize, a collection of randomly selected images from an action class a_n are converted into k 3-way tensors in the latent space-time domain. These k tensors are referred to as the action prototypes for that class.

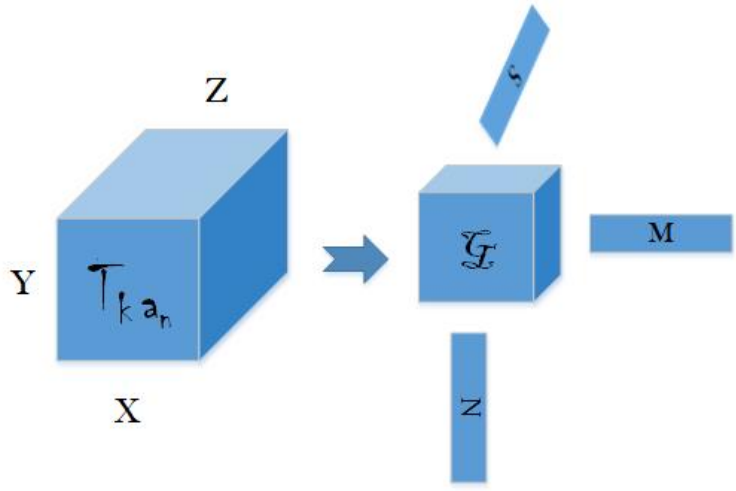

Figure 4.2: Depiction of Tucker decomposition. The tensor $\mathcal{T}_{ka_n} \in \mathbb{R}^{X \times Y \times Z}$ is an action prototype for phase k of an action $a_n \in A$. For rank-1 decomposition $\mathcal{G}$ is a scalar. M, N and S are vectors.

4.2 Rank-1 Subspace Representation of Action Prototypes

Since each action prototype $\mathcal{T}_{ka_n}$ depicts the latent space-time representation of mostly the same phase k of an action a_n, we conjecture that the action prototype tensor $\mathcal{T}_{ka_n}$ can be entirely discriminated by its rank-1 subspace representation, which can be readily computed using a low-rank tensor decomposition method, *e.g. Tucker Decomposition* (*TD*) [121].

Therefore, our next step is to decompose the tensors $\mathcal{T}_{ka_n}$ through *TD* to generate compact signatures for each action prototype group. *TD* is a well-known technique that projects a given tensor $\mathcal{T}_{ka_n} \in \mathbb{R}^{X \times Y \times Z}$ into a smaller core tensor $\mathcal{G}$ and three matrices M, N and S such that

$$\mathcal{T}_{ka_n} \approx \mathcal{G} \times_1 M \times_2 N \times_3 S = \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{z=1}^{Z} g_{xyz} p_x \circ q_y \circ s_z \tag{4.1}$$

where $M \in \mathbb{R}^{X \times K}$, $N \in \mathbb{R}^{Y \times K}$, and $S \in \mathbb{R}^{Z \times K}$ are the orthogonal factor matrices, $\mathcal{G} \in \mathbb{R}^{K \times K \times K}$ is the core tensor and $K \leq min(X, Y, Z)$. The $\overline{\times}_i$ operator denotes the multiplication between a tensor and a vector in mode-i of that tensor, whose result is also a tensor, namely for a tensor $\mathcal{B}$ and vector α the result of their $\overline{\times}_i$ product is a tensor $\mathcal{A}$ given by $\mathcal{A} = \mathcal{B}\overline{\times}_i\alpha \iff (\mathcal{A})_{jk} = \sum_{i=1}^{I} \mathcal{B}_{ijk}\alpha_i$.

Tucker Decomposition of three-way tensors may be viewed as a higher order extension of Principal Component Analysis (PCA) of matrices [122]. It is a rank based estimation, resulting in the decomposition of the tensor in three matrices and one core tensor (see Figure 4.2), where the size of the core tensor is pre-specified. In our case, because of our clustering step, we assume that a rank-1 decomposition would preserve the discriminative properties of the original tensor, *i.e.* K is set to 1, in which case $\mathcal{G}$ reduces to a scalar, and M, N, and S reduce to three vectors with their lengths equal to the image width X, image height Y, and the number of images in the cluster Z, respectively. Intuitively, each of these vectors characterize the intrinsic dynamics of $\mathcal{T}_{ka_n}$ along the

corresponding mode. In particular, each element in $S \in \mathbb{R}^{Z \times K}$ represents the degree of similarity of the corresponding image in the cluster to all others in that cluster. Note that, since the space-time representation of images in one cluster belong to the same action phase, the intra-cluster variance is minimized in each prototype, *i.e.* the variance of the vector S is expected to be small.

Implementation of *TD* is typically iterative, with its computational complexity highly dependent on the strategy used for iterations. Higher Order Orthogonal Iterations (HOOI) [123], a widely-used algorithm for *TD*, is based on Alternating Least Squares (ALS). While the ALS method is not guaranteed to converge to a global optimum, it does converge under certain conditions, in which case it has a local linear convergence rate [124]. Alternatively, differential geometric Newton method could provide local quadratic convergence rate with per iteration cost of $O(H^3D^3)$ for a tensor $\mathcal{X} \in \mathbb{R}^{H \times H \times H}$ and core tensor $\mathcal{G} \in \mathbb{R}^{D \times D \times D}$ [125].

4.3 Action Classification

The last step in the ZTD classifier consists of computing a set of joint probability distributions, $p_{k_n}(a_n, I_t)$, between the class labels a_n and the test image I_t. The unknown class label of the test image is then estimated as the one yielding the maximum joint probability distribution, which by Bayes' law coincides with the maximum *a posteriori* estimate of the unknown label:

$$p(a_t = a_n | I_t) = \arg\max_{k_n, n} \{p_{k_n}(a_n, I_t)\} \tag{4.2}$$

We estimate the joint probability distributions $p_{k_n}(a_n, I_t)$ by the extent a test image I_t affects the intra-class variance of a prototype tensor $\mathcal{T}_{k_n a_n}$, when I_t is added to the prototype tensor. We define the intra-class variance in terms of the rank-1 subspace representation as the variance of the mode-3 vector S, since S characterizes the variations and the dynamics among images within the tensor

[126, 18, 14, 127]. Let $I_1, ..., I_Z$ be the set of images used to construct the prototype tensor $\mathcal{T}_{k_n a_n}$. We then have:

$$
\begin{aligned}
p_{kn}(a_n, I_t) &= p(I_1, ..., I_Z, I_t) && (4.3)\\
&= p(\mathcal{T}_{k_n a_n}, \acute{\mathcal{T}}_{k_n a_n}) && (4.4)\\
&= 1 - \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{D^T D}{2\sigma^2}}, \quad D = S - \acute{S} && (4.5)
\end{aligned}
$$

where $\acute{\mathcal{T}}_{k_n a_n}$ is the prototype tensor $\mathcal{T}_{k_n a_n}$ when a random image in the set $I_1, ..., I_Z$ is replaced with I_t, S and $\acute{S}$ are the corresponding mode-3 vectors of the rank-1 subspace representations of $\mathcal{T}_{k_n a_n}$ and $\acute{\mathcal{T}}_{k_n a_n}$ respectively, and σ is the variance of the residual D. It is worth noting that mode-3 vector is a zero-mean random vector in the range $[-1, 1]$ [126].

4.4 Evaluation

As explained in section 4.2 we form groups of action prototypes by clustering I_Es, columns in matrix $\mathcal{R}$, based on the similarity between their latent space-time representations. Size of images are the same in all action prototype groups. Each group provides evidence for the distinctive compact latent spatiotemporal descriptions of images in that group. Relatively small groups are dropped as they correspond to limited numbers of images, illustrating infrequent action phases in an action class. An image, I_E is pruned out of its action prototype group if the distance of its corresponding entry in S is more than three scaled median absolute deviations away from the local median within a five-element window. Such images fail the condition of latent spatiotemporal similarity with their action prototype group.

We created 325, 123, 25, 134 and 78 action prototype groups for UCFSI-101, *UCF-STAR*, Willow, Stanford-40 and WIDER datasets, respectively. Every test image I_{E_o} has certain conditional

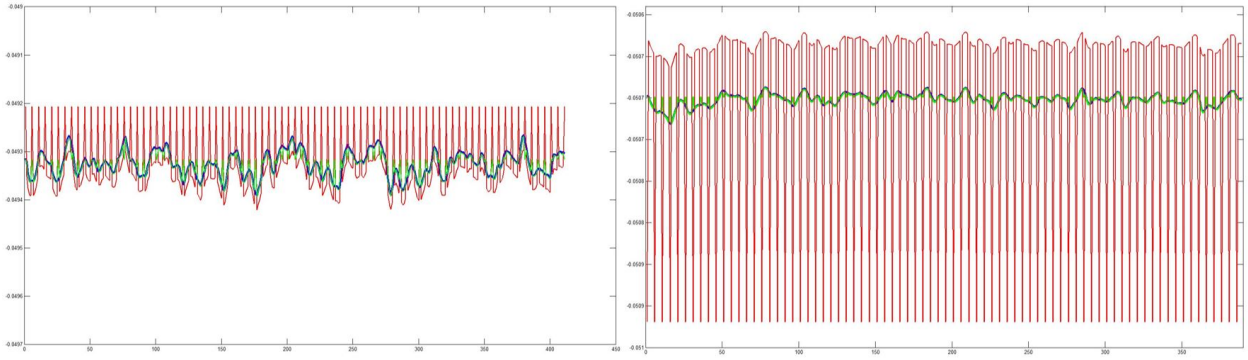

Figure 4.3: Comparison of rank-1 tucker decomposition when similar and dissimilar test image I_t is inserted into an action prototype group; Blue curve: Original signature vector S for an action prototype tensor. Green curve: New signature vector $\acute{S}$ after an image having similar latent spatiotemporal patterns is inserted to the members of the action prototype group. Red curve: New signature vector $\acute{S}$ after an image having different latent spatiotemporal patterns is inserted to the members of the action prototype group.

probability distribution over all $\mathcal{T}_{ka_n}$, denoted by $P(\mathcal{T}_{ka_n}|I_{E_o})$. There is no action label available for I_{E_o}. I_{E_o} is plugged in the tensor $\mathcal{T}_{ka_n}$ at any location h and the vector S is perturbed at and around index h. As shown in Fig. 4.3, the amount of perturbation provides an estimate of how similar/dissimilar the test image is to the group.

Table 4.1: ZTD performance (mAP) on UCFSI-101 by category groups.

	I_{RGB}	$I_{SMPOF_hPOF_v} \in Q$	$I_E \in R$
Human-Object	33.0	0 53.10	62.0
Body-Motion	21.20	78.0	**91.10**
Human-Human	18.02	32.0	59.0
Playing-Instrument	39.23	49.06	68.0
Sport	29.1	64.9	75.86
All group	29.0	57.10	73.20

Our results once again show that appearance and latent motion are complementary sources of information for single image action recognition especially for body-motion group. Table 4.1 shows ZTD's performance on UCFSI-101 images broken down by category groups. We formed three

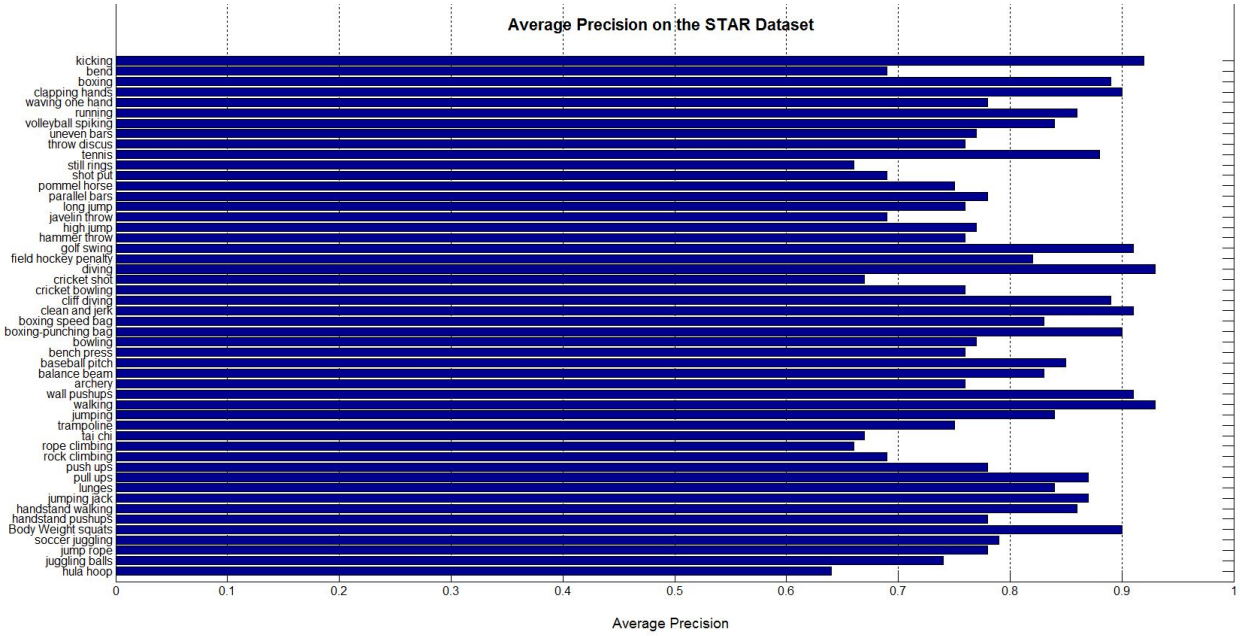

Figure 4.4: Per class performance (mAP) of ZTD on the *UCF-STAR* dataset.

types of action prototypes using RGB images, $I_{SMPOF_h POF_v} \in \mathcal{Q}$ and $I_E s \in \mathcal{R}$ for UCFSI-101 images, separately. Experiments on using RGB images rather than I_Es demonstrate that focusing on latent spatiotemporal pixels' evolution rather than all the pixels in an image improves the results by reducing noise introduced by monitoring all pixels otherwise. Fig. 4.4 represents the action classification accuracy on *UCF-STAR* dataset for each action class. Tables 4.2-4.5 show ZTD's performance on all benchmarks.

It is worth mentioning that other exclusively spatial-based approaches exploit the spatial information. However, they do not help identify which human parts are likely to move. Results on all benchmarks show that our method is not only training-independent, but also yields superior results compared to the state-of-the-art methods by considering the latent spatiotemporal representation of images.

Table 4.2: Results (mAP) on Willow dataset.

Method	**Non-Body-Motion**	**Body-Motion**
Delaitre *et al*. [32]	55.6	62.7
Delaitre *et al*. [76]	55.4	70.7
Sharma *et al*. [109]	59.0	71.1
Sharma *et al*. [81]	60.1	73.2
Khan *et al*. [110]	62.4	72.2
Khan *et al*. [111]	64.1	78.05
Zhao *et al*. [80]	67.8	79.3
Khan *et al*. [42]	62.2	76.0
ZTD	75.3	**81.8**

Table 4.3: Results (mAP) on Stanford-40.

Method	**Body-Motion**	**Non-Body-Motion**	**All**
Gkioxari *et al*. [112]	93.87	89.73	90.9
Khan *et al*. [111]	56.92	51.51	53
Khan *et al*. [42]	53.51	51.28	51.9
Yan *et al*. [113]	92.26	87.07	88.5
Zhao *et al*. [114]	-	-	83.4
Zhao *et al*. [80]	-	-	54.5
Zhao *et al*. [115]	-	-	80.6
Zhou *et al*. [116]	-	-	55.3
Sharma *et al*. [117]	-	-	72.3
Khan *et al*. [118]	-	-	75.4
Gao *et al*. [119]	-	-	74.9
ZTD	**95.5**	78.2	82.9

Table 4.4: Results (mAP) on WIDER.

Method	**mAP**(%)	
RCNN	80.0	
R*CNN	80.5	
DHC	81.3	
ResNet-SRN	86.2	
VeSPA	82.4	
ZTD	Body-Motion	Non-Body-Motion
	88.7	72.7

Table 4.5: Results (mAP) on BU_{101} dataset.

mAP (%)					
Category	Human-Object	Body-Motion	Human-Human	Playing-Instrument	Sport
ZTD	58.9	79.9	59.4	69.4	73.6

CHAPTER 5: TICNN: A HIERARCHICAL DEEP LEARNING FRAMEWORK FOR STILL IMAGE ACTION RECOGNITION USING TEMPORAL IMAGE PREDICTION

In this chapter [1], We first introduce the concept of *temporal Image*I_Ts, using learning parameters of a ranking classifier that sorts the video frames temporally. We then propose a hierarchical deep convolutional neural network, *TICNN*, to (i) design a temporal image prediction model (given a single image), capable of estimating human's action dynamics as a hypothetical sequence of images, (ii) design an action classification model to aggregate predicted temporal image information for human action understanding. Experiments on four challenging datasets demonstrate the importance of generating temporal images in human action prediction, especially when actions rely mainly on human body motions, rather than human-object interactions. Fig. 5.1 illustrates the the *TICNN* architecture.

5.1 Temporal Image

A temporal image, I_T, is a novel compact image representation for a still image I. I_T represents the temporal evolution of pixels within a sequence of images. The concept of I_T was inspired by [102, 128]; *i.e.* modeling video evolution for video action recognition. We introduce a methodology to create I_T for a single frame to serve as its label. We generate an I_T for each frame in the video with respect to a moving window of the neighboring frames. Unlike [128], created I_Ts are used to

[1] This content was reproduced from the following article: Safaei, Marjaneh, Pooyan Balouchian, and Hassan Foroosh. "TICNN: A hierarchical deep learning framework for still image action recognition using temporal image prediction." In 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 3463-3467. IEEE, 2018. This article proudly received the best paper award at "2018 25th IEEE International Conference on Image Processing (ICIP)".

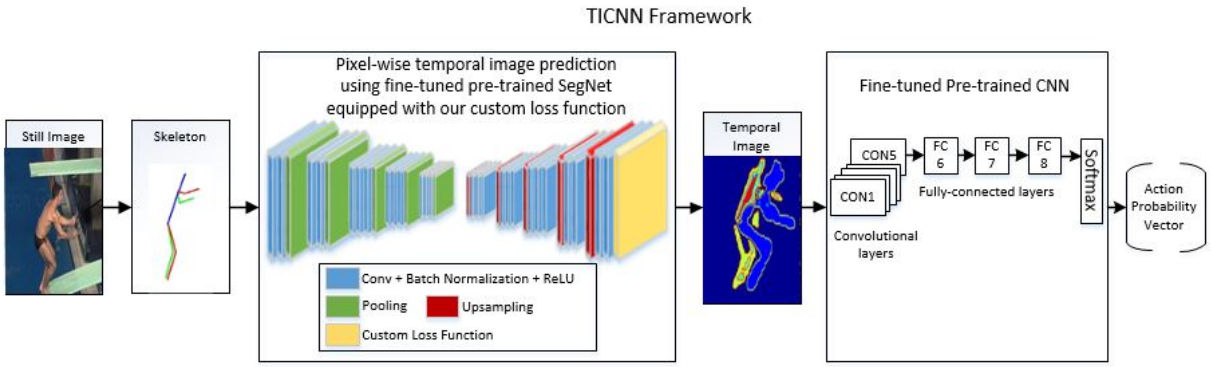

Figure 5.1: Illustration of TICNN architecture. There are two CNNs in a unified hierarchical framework. TICNN first predicts temporal images through pixel-wise image segmentation using a custom loss function developed for this purpose. The predicted temporal images are then used to train an action classification model that predicts an action given a temporal image

learn a prediction model, enabling the prediction of I_T for still images in the wild, lacking motion information.

Generating I_T is essentially done as a ranking process, where the projection kernel is learned from a training set, similar to the domain mapping technique described in section 3.1.2. However, here we apply ranking process over a totally different domain.

The parameters of the linear ranking functions encode the pixel evolution in a principled way. To model such a ranking machine, we use the supervised learning to rank algorithm proposed in [103]. We use the parameters of the ranking machine as a temporal image, I_T, in order to characterize the significance of temporal attributes of each pixel in recognizing an action.

Here, unlike what was described in section 3.1.2, $V = [v_{t_1}, v_{t_2}, ..., v_{t_n}]$ represent a sequence of video frames converted into human body *skeleton* representations, where the frame order also dictates the evolution of the frame appearances. We focus on the relative orderings of the frames. If v_{t+1} succeeds v_t, we have an ordering denoted by $v_{t+1} > v_t$. A linear Rank-SVM represents

a pairwise linear ranking machine that learns a linear mapping of the form $\Psi(\mathbf{V};\mathbf{R}) = \mathbf{R}^T\mathbf{V}$ [103, 104]. We envision the order of the sequence in the training set V as $v_{t_n} > ... > v_{t_2} > v_{t_1}$. The ranking score of v_t is derived by $\Psi(\mathbf{v_t};\mathbf{r}) = \mathbf{r}^T\mathbf{v_t}$ and satisfies the pairwise constraints ($v_{t+1} > v_t$) by a large margin, while avoiding over-fitting.

Consequently, we aim to learn a parametric vector r such that it satisfies all constraints $\forall v_{t_i}, v_{t_j}$, $v_{t_i} > v_{t_j} \iff \Psi(\mathbf{v_{t_i}};\mathbf{r}) = \mathbf{r}^T\mathbf{v_{t_i}} > \Psi(\mathbf{v_{t_j}};\mathbf{r}) = \mathbf{r}^T\mathbf{v_{t_j}}$. The problem of learning the optimal linear kernel for V reduces to solving the following convex optimization problem [103]:

$$\arg\min_{\mathbf{R}} \frac{1}{2}\| \mathbf{R} \|^2 + W \sum_{\forall v_{t_i}, v_{t_j}, v_{t_i} \geq v_{t_j}} \epsilon_{ij} \tag{5.1}$$

$$s.t. \quad \mathbf{R}^T(v_{t_i} - v_{t_j}) \geq 1 - \epsilon_{ij}, \quad \epsilon_{ij} \geq 0, \tag{5.2}$$

where ϵ_{ij} are slack variables and W represents a regularization parameter. Solving this optimization problem leads to learning a vector of parameters R, satisfying the order constraints. As the parameters of R define the frame order of frames V, they represent how the frames evolve with regard to the appearance of the video. Consequently the evolution of pixels is encoded in vector R, used as I_T.

5.2 TICNN Architecture

In this section, we provide details on our hierarchical CNN framework for still image action recognition using temporal image (I_T) prediction.

5.2.1 Temporal Image Prediction

We develop a deep convolutional neural network, trained on still images and their labels; *i.e.* I_Ts, to learn a model, capable of estimating human's action dynamics as a hypothetical sequence of images. Given a still image, this model predicts an I_T. Here, we show that I_T prediction can potentially be achieved through pixel-wise semantic segmentation. We constructed our proposed CNN based on fine-tuning SegNet, a recent contribution to the field in pixel-wise semantic segmentation introduced in [129] as a deep convolutional encoder-decoder neural network architecture.

Unlike SegNet, *TICNN* is not aimed at generating spatial predictions. We, therefore, reconfigured SegNet's classification setup so it fits our model, since *TICNN*'s nature is in conflict with that of SegNet. Next, we implemented a custom loss function, extending Caffe's default *softmax layer* and adjusted the learning rates accordingly. To the best of our knowledge, *TICNN* is the first effort to reuse SegNet to make temporal as opposed to spatial predictions. In SegNet experiments, context is defined as the spatial-relationship between different classes of shapes, while in *TICNN* context is defined as the temporal-relationship between different hypothetical frames (of a non-existing video) in a still image.

SegNet uses a trainable multi-class softmax classifier, classifying pixels independently [129]. Therefore, we first quantize the I_Ts into *C* clusters by *k*-means. The problem is then treated as semantic segmentation, where each pixel is classified as a particular I_T's cluster. The output is further generated as softmax probabilities over the I_T clusters for each pixel. The predicted cluster corresponds to the class with the maximum probability of each pixel. The loss is summed up over all pixels in a mini-batch. It implicitly assumes a uniform pmf for the segmentation classes, prone to noise. We, therefore, developed a custom loss function to minimize the noise by considering only the *k* most-likely clusters; *i.e.* the *k* clusters with the highest probability, and optimized the pre-trained SegNet using our custom loss function in the following equation. Let I represent the

image and Y be the ground truth I_T label represented as quantized clusters. The proposed loss function $L(I, Y)$ is therefore:

$$\hat{L}(I, Y) = -\sum_{i=1}^{M \times N} \sum_{r=1}^{C} \omega_r P_{i,(r)}, \tag{5.3}$$

where ω_r are some weight factors, and

$$P_{i,(r)} = \mathbb{1}(Y_i = (r)) \log F_{i,(r)}(I) \tag{5.4}$$

is the pmf in descending order of values; *i.e.* $P_{i,(1)} \geq P_{i,(2)} \geq ... \geq P_{i,(C)}$. Here, we set k=3 and assume $\omega_r = \frac{1}{K}$ for $P_{i,(1)}, \ldots, P_{i,(K)}$, and ω_r=0 otherwise. This is equivalent to averaging over the probabilities of the k most-likely clusters. The $F_{\mathrm{i,r}}(I)$ represents the probability that the i_{th} pixel belong to cluster r, and $\mathbb{1}(Y_{\mathrm{i}} = r)$ is an indicator function.

A predicted I_T represents a compact representation of hypothetical sequence of images of a non-existing video. These temporal patterns are produced as an auxiliary form of information for still image action recognition. In Fig. 5.2 some example of predicted temporal images are depicted. In section 5.2.2 we explain how we benefit from the predicted I_Ts in still image human action classification.

5.2.2 Action Classification

TICNN contains a second CNN, trained on the I_Ts generated by the first CNN, to capture the I_T features and classify actions. Our classifier is similar to the standard 7-layer architecture proposed in [130]. This network is formed by 5 successive convolutional layers followed by 3 fully-connected layers. The 3 fully-connected layers compute $Y_6 = \sigma(W_6 Y_5 + B_6)$, $Y_7 = \sigma(W_7 Y_6 + B_7)$, $Y_8 = \psi(W_8 Y_7 + B_8)$, where Y_{m} denotes the output of the m_{th} layer and W_{m} and B_{m} are

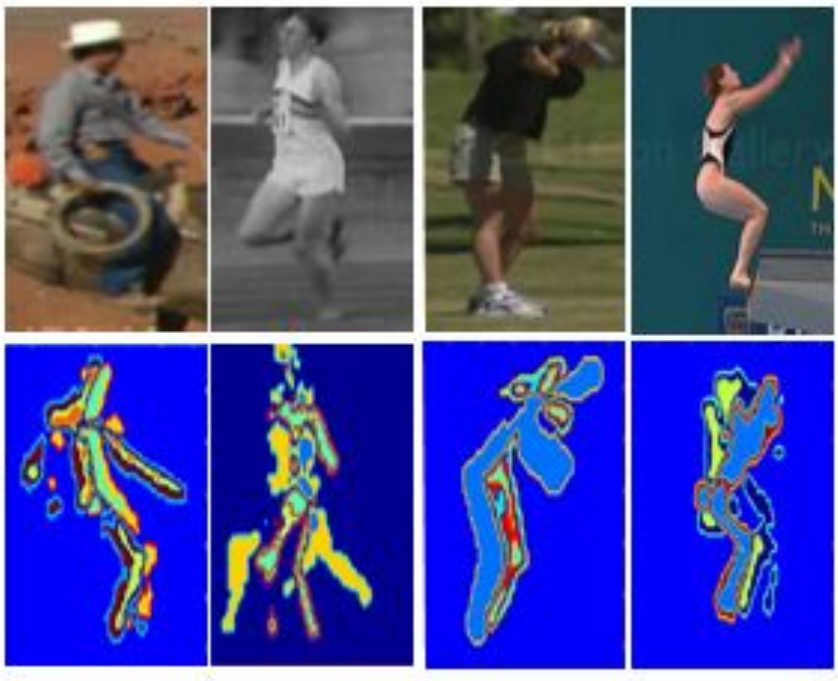

Figure 5.2: Excerpts from predicted Temporal Images by TICNN.

trainable parameters of the m_{th} layer. Also, $\sigma(X)[i] = max(0, X[i])$ and $\psi(X)[i] = e^{X[i]}/\Sigma_j e^{X[j]}$ are the "ReLU" and "SoftMax" non-linear activation functions.

We changed *FC8* layer to enable adapting it to the new target domain, I_Ts. Finally, the new *FC8* is trained on the target dataset from scratch, using a higher learning rate. Our goal was to learn a mapping between $I_T s$ and the action labels. All the strides are set to 1 except for the first layer, set to 4 during convolution. The stride for pooling is 2, and we configure the pooling kernel size as 3×3. A base learning rate of 0.001 and a step size of 70,000 iterations were configured. Here, we perform a supervised domain adaptation via fine-tuning a network specifically pre-trained for appearance-based classification on RGB images, later used to train a network for action-based classification on I_Ts domain. We evaluate the proposed convolutional network in section 5.3.

5.3 Evaluation

Here we provide details on the datasets we used as well as the experiments ran to perform (1) temporal image prediction, and (2) still image action classification.

5.3.1 Datasets

UCF Sport-SI. We first generated a large annotated still-image dataset, by extracting over 10,000 frames from *UCF Sport* video dataset [131] of 150 videos and 10 action categories including Diving, Golf-swing, Kicking, Lifting, Riding-horse, Running, Skate-boarding, Swing-bench, Swing-side and Walking.

We used the authors' original train/test splits. The extracted frames, after the sampling process, were post-processed to eliminate frames with no clearly visible human objects. The data in the dataset were then augmented by flipping images, resulting in a dataset consisting of 18,000 frames split into 12,000 training images, labeled based on the video labels they belong to, and 6,000 test images. This dataset is utilized in producing the I_Ts used to train *TICNN*'s first CNN.

Google and iStockPhoto. We formed a dataset that plays a key role in our experiments using google and istockphoto images , representing 10 action categories including Diving, Golf-swing, Kicking, Lifting, Riding-horse, Running, Skate-boarding, Swing-bench, Swing-side and Walking. This dataset contains 1000 images, with a 700 and 300 train/test split, representing 10 action categories, with the human body located in the center of the image. All these images are further used for testing out the models trained in STCNN. The reason why this dataset was formed is to support the claim that STCNN is capable of performing well even when given images, belonging to the mentioned categories, are downloaded from the Web.

We also used the *Willow* action dataset, *Stanford-40* as well as a newly collected dataset using *Google* and *iStockPhoto* images (supporting the claim that *TICNN* performs well when tested on images from the wild) with all 10 action categories in *UCF Sport*, containing 1000 images with a 700/300 train/test split. *TICNN* uses these images to further test the trained models.

5.3.2 Temporal Image Prediction Training and Analysis

RGB frames from *UCF Sport-SI* were converted to the human body skeleton representation using Stacked Hourglass Networks [132]. Skeleton is a lower dimensional shape description of an object. Hence, I_Ts generated from human skeletons carry less noise. Using the algorithm discussed in section 5.1, I_Ts were generated from the skeleton images, forming *UCF Sport-TI* dataset. Some examples of the generated I_Ts are shown in figure 5.3. *UCF Sport-TI* is used in training *TICNN*'s first CNN to learn a model for predicting I_Ts.

UCF Sport TI is used to train SegNet from scratch. The challenge is to segment I_Ts into *k*=10 classes, using *k*-means clustering. Each class represents the intensity value of the relevant pixel in the I_T. SegNet has an extra class, representing unclassified pixels. *TICNN*'s classification nature, however, is existential since it is not concerned with object detection or scene understanding. It rather attempts to detect if a pixel carries any valuable information; *i.e*. a pixel, part of the object in the image. A pixel carrying information then gets classified as one of the existing *k* classes based on its intensity value. Therefore, we eliminated the ignore label construct.

We chose a mini-batch size of 6 and iterations of 100,000, 400,000 and 1,000,000, resulting in 50, 200 and 500 epochs respectively. We monitored the validation error during the training process to ensure the increased number of epochs won't overfit the model. We chose the model yielding the best results to generate I_Ts, fed into our second CNN, tasked with still image action classification. We ran our experiments on a GeForce GTX TITAN X machine with 15 GB memory and a second

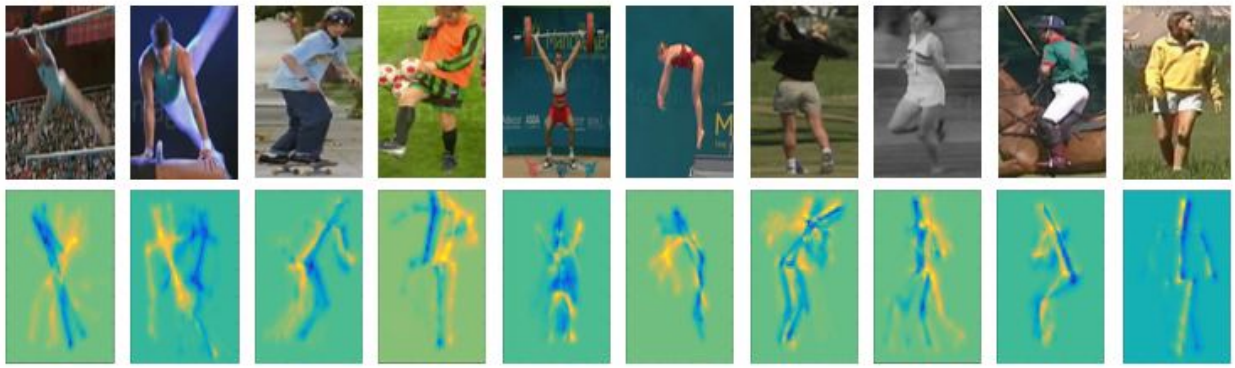

Figure 5.3: Excerpts from Temporal Images generated from UCF Sport dataset.

NVIDIA Tesla M60 with 16 GB of memory.

We initialized the encoder weights from VGG model trained on ImageNet and ran it on CamVid [133]. This model was then used to fine-tune our CNN on *UCF Sport TI*. Table 5.1 shows an increase in the accuracy from 44% to 49%, 56% to 63% and 65% to 74% under 50, 200 an 500 epochs respectively. We also show the effect of utilizing the custom loss function and adjusted the learning rates. Table 5.1 shows a noticeable boost in accuracy compared to experiments run with the default softmax loss layer. We show an increase in accuracy from 74% to 82% using this implementation and configuration.

Table 5.1: Quantitative comparisons based on different epochs.

Model	**50 Epochs**	**200 Epochs**	**500 Epochs**
Training SegNet from scratch	44%	56%	65%
Fine-tuned SegNet using VGG model	49%	63%	74%
Fine-tuned with custom loss function	55%	70%	**82%**

5.3.3 Action classification training and analysis

In this section, we explore the training of our still image action classification model. We first used *UCF Sport-TI*, containing the I_Ts along with their action categories, to train our second CNN. Here, we propose a transfer learning method by fine-tuning a network pre-trained on a different source domain, over our predicted I_Ts. Our transfer learning method yields promising results when compared against training the CNN from scratch on our predicted I_Ts, as shown in table 5.2.

Next, we use the model derived from section 5.3.2 to produce I_Ts for the body motion category of *Willow*, our cherry-picked *Google-iStockPhoto* and *Stanford-40* datasets. Having labeled I_Ts, we perform action classification on I_T images. We, therefore, removed the last fully-connected layer from our network, pre-trained on *UCF Sport-TI*. Next, we trained a linear softmax classifier for the I_Ts generated from all other datasets. Tables 5.2- 5.5 show action classification results accordingly, demonstrating the positive effect of using I_Ts in still image action classification.

Unlike conventional methods, we treat actions with similar poses, *e.g.* *running* vs. *walking*, differently, benefiting from the presence of temporal information in I_Ts, carrying information on the pixels evolution. Even though our I_T prediction model was not trained for Bike-riding, a reasonable accuracy was achieved with reference to table 5.5. The proposed methods in [109, 82, 81] have mainly relied only on body parts and poses as cues for action recognition. The extensive manual labeling of human bounding boxes and having enough number of body pose and part detectors, combined with the fact that different actions might have very similar poses, introduce some challenges. As experiments in tables 5.2- 5.5 show, our method, by benefiting from I_Ts rather than relying only on spatial patterns, reports superior results.

It is worth mentioning that, the motivation behind this work is absence of temporal information in still images that poses a serious challenge in still image action classification. We designed a CNN

Table 5.2: Comparison of action classification accuracy run on UCF Sport TI, Google and iStock-Photo datasets

Dataset	Model	Dive	Golf	Kick	Lift	Walk	Skate	Swing-side	Swing-bench	Horse-ride	Run	All
UCF Sport TI	Trained from scratch	0.67	0.65	0.48	0.63	0.49	0.43	0.48	0.39	0.57	0.52	0.53
UCF Sport TI	Fine-tuned all layers	0.75	0.72	0.55	0.74	0.57	0.55	0.59	0.47	0.62	0.61	0.62
UCF Sport TI	Fine-tuned 5 top layers	0.88	0.85	0.76	0.87	0.72	0.74	0.73	0.69	0.86	0.84	0.79
Google and iStockPhoto	Model learned from UCF Sport TI	0.82	0.86	0.67	0.85	0.66	0.65	0.73	0.64	0.81	0.82	0.75

Table 5.3: Stanford-40 results with per-action accuracy

Method	Gkioxari *et al.* 2016	Yan *et al.* 2017	Khan *et al.* 2014	Khan *et al.* 2013	Ours
Body-motion	93.87	92.26	56.92	53.51	**96.8**
Non-body motion	89.73	87.07	51.51	51.28	75.0
All	90.9	88.5	53	51.9	80.9

that uses the capabilities of pixel-wise image segmentation to learn a model that predicts a temporal image given a still image. We then fine-tuned a second CNN to perform action classification based on the predicted temporal images. We demonstrate that *TICNN*, to the best of our knowledge and for the first time in the literature, eliminates the need for having videos to generate temporal images. We inject temporal information to still images to enable treating temporal information as context, while related efforts mainly consider spatial information as the context. We carried on a comprehensive analysis on all components of our proposed architecture, and compared our extensive experimental results against state-of-the-art, showing that *TICNN*'s results are superior.

Table 5.4: Stanford-40 results without per-action accuracy

Method	Zhao et.al 2017	Zhao et.al 2017	Zhao *et al.* 2017	Zhou *et al.* 2014	Sharma *et al.* 2015	Khan *et al.* 2015	Ours
Body-motion	-	-	-	**-**	-	-	**96.8**
Non-body motion	-	-	-	-	-	-	75.0
All	83.4	54.5	80.6	55.3	72.3	75.4	80.9

Table 5.5: Comparison (mAP) on the Willow dataset.

Method	**Bike-ride**	**Horse-ride**	**Run**	**Walk**	**Overall (Body-Motion)**
Delaitre *et al.* [32]	82.43	69.60	44.53	54.18	62.7
Delaitre *et al.* [76]	90.39	75.03	59.73	57.64	70.7
Sharma *et al.* [109]	87.8	84.2	56.1	56.5	71.1
Sharma *et al.* [81]	91.0	87.6	55.0	59.2	73.2
Khan *et al.* [110]	87.2	77.2	63.7	60.6	72.2
Khan *et al.* [111]	93.8	87.9	67.2	63.3	78.05
Liang *et al.* [82]	98.17	92.72	46.16	58.88	74.0
Zhao *et al.* [80]	93.0	86.2	65.7	72.6	79.3
Khan *et al.* [42]	90.3	84.3	64.7	64.6	76.0
TICNN	79.8	87.2	81.7	80.4	**82.3**

CHAPTER 6: UCF-STAR: A LARGE SCALE STILL IMAGE DATASET FOR UNDERSTANDING HUMAN ACTIONS

Action recognition in still images poses a more serious challenge particularly due to the fewer available training data. To address this challenge, we introduce an image a dataset for STil image Action Recognition (STAR) [1], containing 1,038,622 annotated still images, collected from the *wild*. To the best of our knowledge, *UCF-STAR* is the largest dataset in the literature, more than 40 times the size of the largest previous action image dataset; *i.e*. BU-101 [29], meant for action recognition in still images. The key characteristics of *UCF-STAR* include: (1) focusing on the human body-motion rather than action categories involving relatively stationary human-object interaction, (2) making use of images collected from the wild to benefit from a varied set of action representations, (3) providing multiple human-annotated labels rather than just the action label, and (4) rich, structured and multi-modal set of metadata for each image. This departs from existing datasets for still image action recognition, which typically provide sparse annotations in less number of images and categories. *UCF-STAR* exposes the intrinsic difficulty of action recognition through its realistic scene and action complexity. Excerpts from *UCF-STAR* dataset are shown in Fig. 6.1.

From a temporal granularity perspective, human actions fall into two categories; *i.e*. (1) human body-motion, and (2) stationary actions. While in existing datasets many action classes, *e.g*. *photography*, are relatively static and dependent on human-object interaction, our emphasis is on human body-motion actions, *i.e*. actions dependent on body motion. Thus, *UCF-STAR* is constructed by collecting images of actions with body motion, and is fully benchmarked in section 6.4. On the other

[1]This content was reproduced from the following article: Safaei, Marjaneh, Pooyan Balouchian, and Hassan Foroosh. January 2020 Conference: Thirty-Fourth AAAI Conference on Artificial Intelligence- AAAI 2020At: New York, USA

Figure 6.1: Excerpts from *UCF-STAR*: (a) Examples depicting body-motion actions; (b) Examples of associated metadata and labels, *e.g.* bounding boxes, action class, captions, tags, number of humans, human visibility and human-object interaction.

hand, actions could be treated as person-centric (individual), or group activities. Our focus is on the actions performed by people, treated as individual agents. There can be multiple people in a scene, however each one serves as an individual agent. Here, the annotations are constructed to refer to the main agent.

UCF-STAR contains still images focusing on person-centric body-motion actions. Hence, the choice of keywords to crawl the Web was driven by the mentioned concepts. For instance, *swinging tennis racket* took precedence over simply *tennis*. We formed 50 different action categories, searched the Web via the Bing's Cognitive Services API [134], and enriched the dataset with metadata including human visibility, number of humans, human-object interaction, captions, tags, bounding boxes, action labels, among more metadata further explained in the next section. Table 6.1 compares *UCF-STAR* against some well-known image datasets for action recognition with respect to the main dataset statistics.

Table 6.1: Comparison of *UCF-STAR* dataset with other still image action recognition datasets.

Dataset	#classes	#images	Labels			Caption	Source	Tag	Bounding box
			number of humans?	human-object interaction?	human visibility?				
Stanford-40 (Yao *et al.*)	40	9,532	No	No	No	No	No	No	Yes
Willow (Delaitre *et al.*)	7	911	No	No	No	No	No	No	Yes
Pascal VOC 2010 (Maji *et al.*)	9	50 to 100 per class	No	No	No	No	No	No	Yes
Pascal VOC 2011 (Maji *et al.*)	10	200 or more per class	No	No	No	No	No	No	Yes
Pascal VOC 2012 (Maji *et al.*)	10	200 or more per class	No	No	No	No	No	No	Yes
PPMI (Yao and Fei-Fei)	7	2,100	No	No	No	No	No	No	No
89 Action Dataset (Le *et al.*)	89	2,038	No	No	No	No	No	No	No
BU-101 (Ma *et al.*)	100	23,782	No	No	No	No	No	No	No
Action Images by Ikizler (Ikizler *et al.*)	6	467	No	No	No	**Yes**	No	No	No
Sport Dataset (Gupta *et al.*)	6	300	No	No	No	**Yes**	No	No	No
UCF-STAR	**50**	**1,038,622**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

6.1 UCF-STAR Construction

Construction of *UCF-STAR* was a five step process involving (1) action category selection, (2) semantic grounding, (3) collecting images from the *wild*, (4) image annotation, and (5) enhancing dataset size. Below, we provide the details.

6.1.1 Action Category Selection

We followed two principles in selecting action categories. First, only actions involving significant human body-motion were selected. Second, key poses providing clear visual signatures were considered for each action; *i.e*. *tennis swing* or *tennis serve* taking precedence over *playing tennis*. Needless to emphasize on the generic nature of the term *playing tennis* compared to *tennis swing/serve*.

6.1.2 Semantic Grounding

To search for images, we applied semantic grounding to find synonymous terms for actions, leading to more accurate data retrieval. This was done by querying *WordNet* synsets and collecting synonymous terms. This step was performed to not only expand the search space, but also help

retrieve more relevant images, minimizing search misses of conventional keyword-only searches. To retrieve human-centric images, we appended keywords like ”human”, ”person”, ”woman” or ”man” to form *n-grams*; *i.e.* human + <action>, person + <action>, man + <action> and woman + <action>. This leads to a significant reduction in noise that would otherwise include images with no visible human body.

6.1.3 Image Collection

To crawl the *Web* for images, we took advantage of *Bing's Cognitive Services API* as proposed in [134], supporting 250 transactions per second, where a transaction is defined as a successful Bing API call request. This API provides support for an array of filters including *face-only*, *include body parts*, etc., as part of its *Image Search API*. We flagged each *n-gram* with the relevant filters, such as *face-only* and *include body parts*. These flags help the final search results require less manual effort, and reduce noise. Using this approach, we collected 29,037 images, which we refer to as the *strongly labeled* dataset. Even though a dataset of 29,037 images would be considered as the largest action image dataset in the literature, we further enhanced the dataset size using the approach explained next.

The images in each class have large variations in background, appearance and pose. To further enhance *UCF-STAR*, we also collected a rich set of *metadata* for each image. Bing Image Search API returns *insightsToken* that can be used to submit a second query for collecting a rich set of metadata on each image, including: 1) *BRQ* which is the best representative query that is defined as a term that best describes the image, 2) *Caption*, which provides textual information that may contain entities and links to other related entities, 3) *Collections* providing a list of related images, 4) *PagesIncluding* providing a list of webpages that include the image, 5) *RecognizedEntities* representing a list of entities (people) that were recognized in the image, 6) *Re-*

latedSearches offering a list of related searches made by others, 7) *SimilarImages* providing a list of images that are visually similar, and 8) *Tags* providing characteristics of the type of content found in the image. For example, if the image is of a person, the tags may indicate gender or type of clothes they are wearing. *UCF-STAR* is publicly available including all metadata at `https://cil.cs.ucf.edu/dataset-2/ucf-star-2/`

6.1.4 Image Annotation Process

To annotate the collected images, Amazon Mechanical Turk (AMT) workers were employed, answering a number of questions designed to capture the (1) observed human action, (2) number of humans in the image, if any, (3) whether or not there exists at least one whole human body in the image, and (4) whether or not any human-object interaction is observed in the image. Fig. 6.2 depicts the user interface designed as part of the annotation process on AMT. As observed in this figure, the questions are designed in a simple form to minimize any chance of misunderstanding, while at the same time noticeable amount of information is collected.

In practice, it is inevitable for AMT workers to introduce noise, therefore we asked multiple annotators to answer each question. Each image is annotated by three independent AMT workers. We only regard a label as ground truth if it is verified by at least two annotators. Our AMT workers flagged 57.7% of the images as correct; *i.e.* the image matched the weak label (the human action keyword used during search) it initiated from, eventually resulting in 16,756 noise-reduced strongly labeled images enriched with *action* labels and answers to the designed questions.

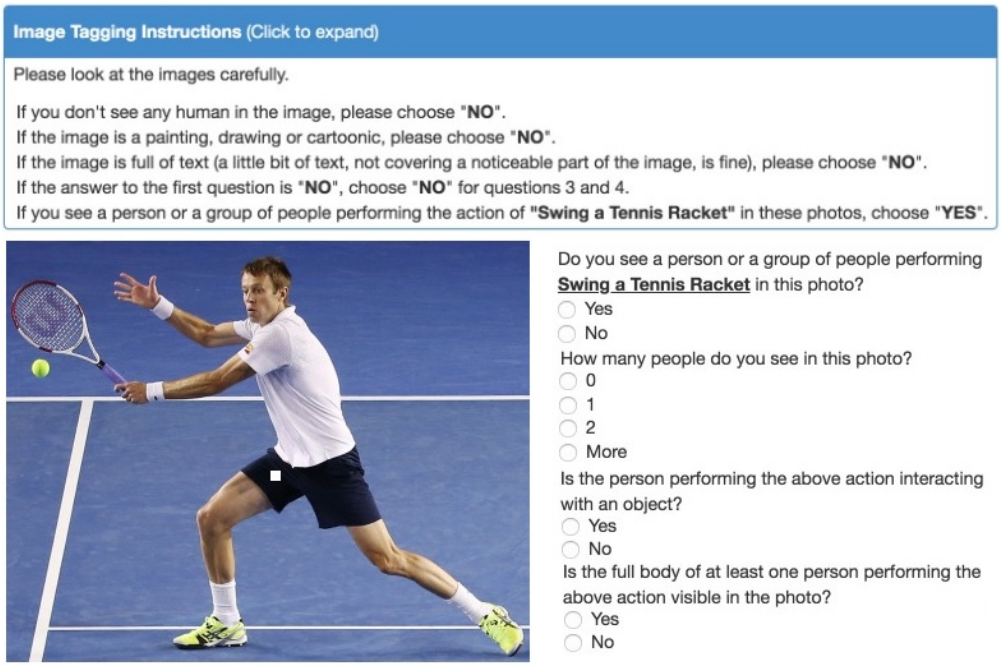

Figure 6.2: User interface for image annotation.

6.1.5 Enhancing Dataset Size

Our dataset, at this step of the process, included only the noise-reduced images labeled by AMT workers. To enhance the size of our dataset, we took advantage of Bing's feature available in its *Image Search API* as proposed in [134] that enables queries for visually similar images. Taking advantage of this feature, our system re-crawled the *Web* and collected 1,315,714 images. Next we removed the duplicates using *fdupes*, resulting in 1,038,622 unique labeled images. We further split the 1,038,622 images into mutually exclusive 664,718 training, 166,180 validation, and 207,724 test images.

6.2 Dataset Statistics

A key characteristic of *UCF-STAR* is its human body-motion based action classes. The rich set of metadata offered by UCF-*STAR* enables the computer vision community to benefit from its multi-modal nature, benchmarking methods relying on the structured metadata contained in the dataset. Fig. 6.3 shows UCF-*STAR*'s distribution of action classes as well as the annotations thereof. Even though the number of images per class is different (averaging at 20,366), the large scale nature of the dataset however enables us to easily sub-sample the dataset to avoid the *class imbalance* problem. Fig. 6.4 depicts the size of action class in the UCF-*STAR*.

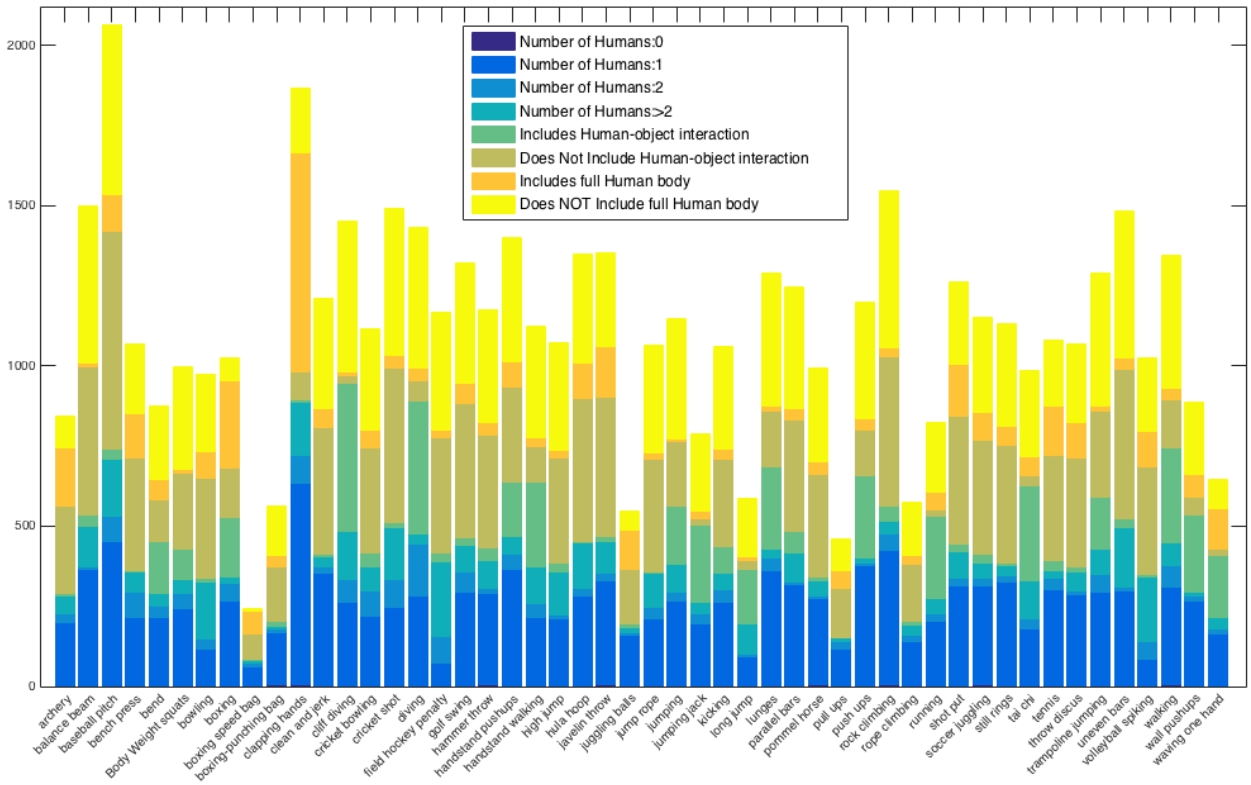

Figure 6.3: Distribution of UCF-*STAR* images' annotations per class.

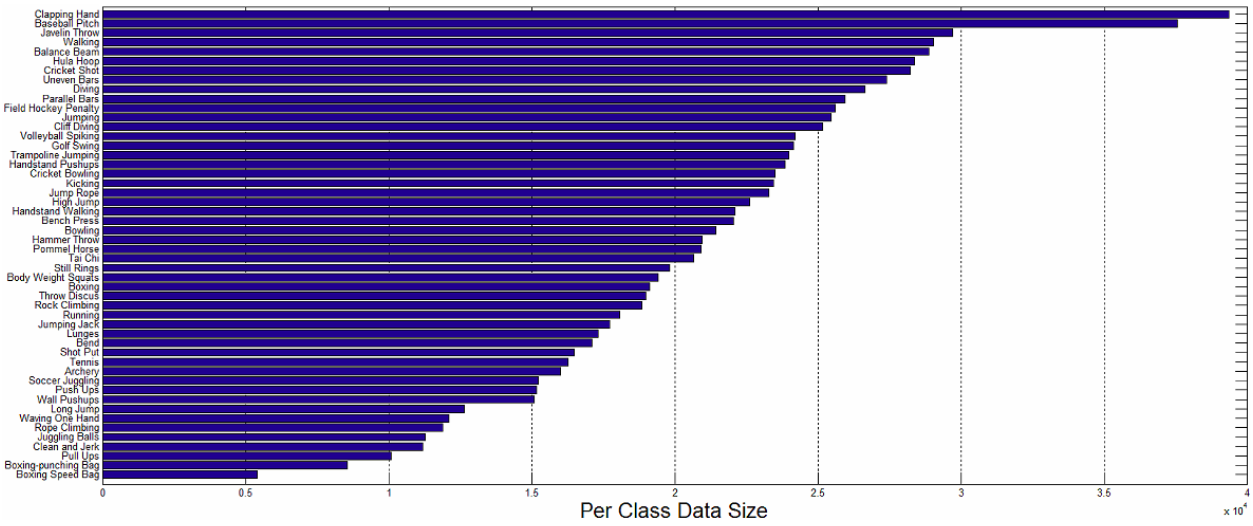

Figure 6.4: Sizes of each action class in the UCF-*STAR* dataset sorted by descending order

6.3 TSSTN Action Recognition Method

While in videos one can readily infer motion [21, 22, 23, 24], such information is missing in a single image. Thus, action recognition poses a bigger challenge in still images, due to the absence of temporal information [80, 27, 39, 40, 135, 42]. The issue is exacerbated when there is no contextual information, *e.g.* interaction with a recognizable object. To address this gap and hence boost the accuracy, we propose a new method of modeling the "latent" temporal information in a still image, and use it as prior knowledge in a two-stream deep network [136, 137, 138, 139, 135].

Popular datasets such as Standford-40, PASCAL VOC and Willow have been widely used by recent methods in still image action recognition. However, the small number of action classes, limited number of images in each class, and the distinctive nature of action categories may present an exaggerated picture of the state of the art. Difficulties arise when the number of classes are large, human-object interaction is not a determining factor in recognition, actions are only subtly different

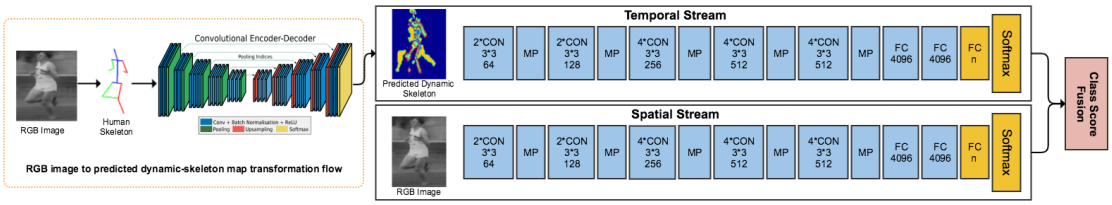

Figure 6.5: Two-stream still image action recognition network, using predicted dynamic-skeleton map as input to temporal stream.

in poses, and background scenes are not informative. *UCF-STAR* has all these aspects aplenty. To prove this, we developed a new action recognition method inspired by recent motion prediction approaches, and compared the results with recent state-of-the-art still image action recognition methods on both *UCF-STAR* and other existing popular datasets mentioned above.

Unlike previous efforts, we propose a new method of modeling the *latent* temporal information in a still image, and use it as prior knowledge in a two-stream deep network, rather than relying solely on spatial information. The key idea is to transfer the temporal information learned from video frames into still images to aid action recognition. To achieve this, we developed a two-stream spatiotemporal network (TSSTN), similar to networks used in the video literature [136, 137, 138, 139, 135], and decomposed still image action recognition into spatial and predicted temporal streams as described below. Fig. 6.5 illustrates the overall architecture of the two-stream network.

Temporal stream network. Our goal is to derive a new image representation, named *dynamic-skeleton map* (d_s), earlier introduced as *temporal image* in section 5.1, by learning motion from video frames and then transferring it to still images. Therefore, a dynamic-skeleton map serves to model the missing temporal information. Dynamic-skeleton represents motions of human body pixels in a predefined time window, hallucinating the human body motion.

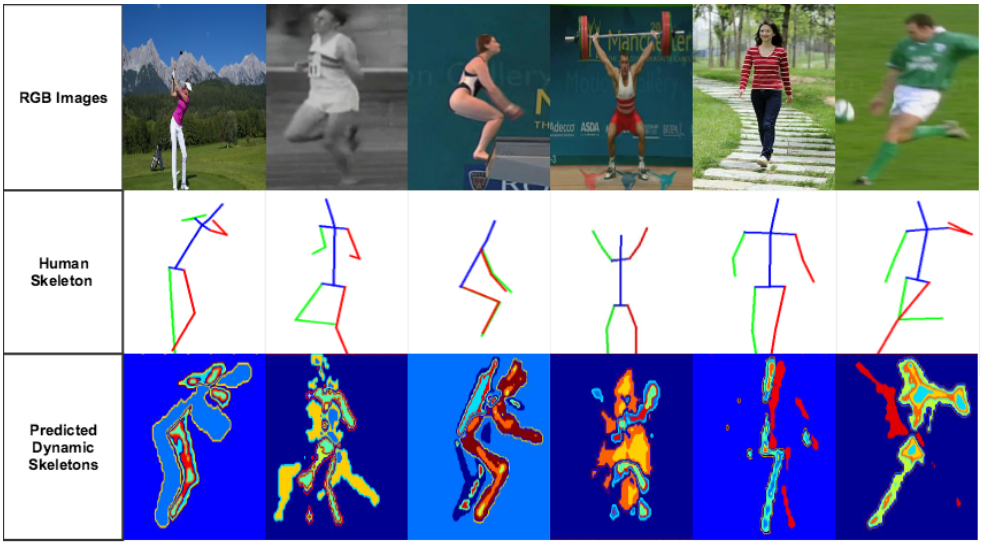

Figure 6.6: Mapping examples from RGB to skeleton, and then to predicted dynamic-skeleton domain.

The concept of dynamic image; *i.e*. modeling video evolution for action recognition, is inspired by [102, 128]. While they propose methods to capture video-wide temporal information for action recognition, we generate a dynamic-skeleton map for every frame in a video to serve as the temporal label for that frame. We then use these labeled frames to learn a model for predicting the dynamic-skeleton maps of still images. We fully elaborated the concept of dynamic-skeleton (earlier introduced as temporal images) and the methodology to predict the dynamic-skeleton map for a single image, when video data is not available,in section 5.1 and 5.2.1, respectively.

A predicted dynamic-skeleton, d_s, represents a compact temporal representation of a a still image, as a hallucinated sequence of human skeleton images. These temporal patterns can then be used as auxiliary information for action recognition in still images. Fig. 6.6 depicts examples of predicted dynamic-skeleton maps for some still images. Predicted d_ss are used as the input to the temporal stream in our two-stream action recognition network depicted in Fig. 6.5.

Spatial stream network. The spatial stream network operates on individual RGB images, performing action recognition from still images. The static appearance by itself is a useful cue, and improves the final classification score. Since the spatial stream is essentially an image classification network, we can build upon the recent advances in large-scale image recognition methods.

We build TSSTN using a CNN architecture similar to [107]. Our temporal and spatial streams are trained to selectively focus on the corresponding features, respectively. Each stream is formed by sixteen successive convolutional layers followed by three fully connected layers. We denote the convolutional layers as CON(*k*,*s*), indicating that there are *k* kernels, of size $s \times s$. The input to our CNN is a fixed-size 224×224 image. The convolution stride is fixed to 1 pixel. Max-pooling is performed over a 2×2 pixel window, with stride 2. Finally, FC(*n*) denotes a fully connected layer with *n* neurons. We change the last *FC* layer, used smaller learning rates for layers that are being fine-tuned to further promote our goal to inject the predicted motion prior into still images.

Final classification score. Stacked generalization is a method of using a high-level model to combine lower-level models to achieve greater accuracy. Stacking with Multi-response regression (MRR) uses linear regression to perform classification [140]. If the original classification problem has I classes, it is converted into I separate regression problems.

Given a data set $\mathcal{D} = \{(y_n, x_n), n = 1, ..., N\}$, where y_n is the class value and x_n is a vector representing the attribute values of the n-th instances, randomly split the data into J almost equal parts, the linear regression for class c is simply:

$$\mathcal{R}_c(x) = \sum_{k}^{K} \alpha_{kc} \mathcal{P}_{kc}(x) \tag{6.1}$$

let $\mathcal{P}_{kc}(x)$ denote the probability of the c_{th} output class obtained by the k_{th} model for an instance

x. Next, we choose the linear regression coefficients $\{\alpha_{kc}\}$ to minimize

$$\sum_{d} \sum_{(y_n,x_n)\in\mathcal{D}_j} (y_n - \sum_{k} \alpha_{kc} \mathcal{P}_{kc}^{-j}(x_n))^2 \tag{6.2}$$

A least-square algorithm under non-negativity constraint is then employed to derive the linear regression for each action class. Finally, to classify a new instance x, $\mathcal{R}_c(x)$ for all C classes is computed and the instance x is assigned to that class c with the greatest value:

$$\mathcal{R}_c(x) > \mathcal{R}_{\acute{c}}(x) \quad for \quad all \quad c \neq \acute{c} \tag{6.3}$$

Since diversity is relatively high among the two classifiers before fusion, simply averaging fusion produces poor results compared to the MLR fusion.

6.4 Experiments

In this section, we experimentally analyze the key features of *UCF-STAR* and the challenges it introduces. As shown in Fig. 6.3, in *UCF-STAR* all classes are of sufficient and roughly equal size, therefore there are no issues of unbalanced classes. Our resulting benchmark consists of a total of 664,718 training, 166,180 validation and 207,724 test examples on 50 classes. We fully compare *UCF-STAR* with existing image datasets in terms of their challenges. We used the train and test splits provided by the original authors for all datasets. For evaluation, we compute the average precision per class and report the average over all classes.

6.4.1 Dynamic-skeleton prediction and analysis

In order to learn the latent motion prior from video frames, we extracted over 36,000 frames from *UCF-101* [96], *UCF-Sport* [131], *WEIZMANN* [141], *KTH* [142] video datasets categorized in 50 different action classes. These extracted frames, after the sampling process, were post-processed to eliminate frames with no clearly-visible human subject. The extracted frames were also augmented by flipping images, resulting in 57,600 frames. We labeled these frames with the video action they were sampled from. Next, RGB frames were converted to human body skeleton representation using Stacked Hourglass Networks [132]. Skeleton is a lower dimensional shape description of an object. Consequently, the domain of inferred motion, based on human skeletons, helps to get rid of irrelevant information, and mitigate over-fitting.

Since the video data are available for the 57,600 extracted frames, we generate d_s labels for them as described earlier in section 5.1. Generated d_ss, using the Rank-SVM algorithm, form our labels for training a model to predict d_s for a still image with no video data available. As discussed in section 5.2.1, we devise a pixel-wise semantic segmentation encoder-decoder architecture for d_s prediction.

6.4.2 Comparison to the state of the art

Tables 6.2-6.5 show action recognition performance of the proposed TSSTN method on *UCF-STAR*, as well as 4 other standard image datasets. TSSTN obtains state-of-the-art performance on Stanford-40, Willow, WIDER and BU-101, outperforming well-established baselines. However, table 6.2 shows that models in [143, 144, 135, 119] obtain relatively low performance on *UCF-STAR*. We attribute this to 1) the human body-motion characteristics of *UCF-STAR*, and 2) existence of visually similar poses performing different actions. Therefore, rich temporal predic-

tion models may be needed to succeed at *UCF-STAR*, posing a new challenge for visual action recognition.

A very key observation is that unlike conventional action classification methods, TSSTN treats actions with similar poses, *e.g*. *running* vs. *walking*, differently. This is due to presence of temporal information in dynamic-skeletons carrying information on the pixels evolution, which would otherwise be missing. Our promising performance on body-motion categories in tables 6.3-6.5 shows the impact of the temporal prediction models in our action recognition method.

Table 6.2: Action classification performance on *UCF-STAR*.

Method	**mAP(%)**
Object Bank [143]	26.7
LLC [144]	31.5
R*CNN [135]	65.3
im2flow [119]	70.9
Temporal Stream-Trained from scratch	61.3
Temporal Stream-Fine-tuned all layers	68.3
Temporal Stream-Fine-tuned 7 top layers	**86.2**
Spatial Stream-Fine-tuned 7 top layers	**26.3**
TSSTN	**91.9**

TSSTN proves that predicting the latent temporal information in still images improves action recognition performance. Moreover, *UCF-STAR* highlights the need for developing new action recognition approaches based on predicting temporal information in still images.

Table 6.3: mAP(%) results on Stanford-40.

Method	Body-Motion	Non-Body-Motion	All
Gkioxari *et al*. [135]	93.87	89.73	90.9
Khan *et al*. [111]	56.92	51.51	53
Khan *et al*. [42]	53.51	51.28	51.9
Yan *et al*. [113]	92.26	87.07	88.5
Zhao *et al*. [114]	-	-	83.4
Zhao *et al*. [80]	-	-	54.5
Zhao *et al*. [115]	-	-	80.6
Zhou *et al*. [116]	-	-	55.3
Sharma *et al*. [117]	-	-	72.3
Khan *et al*. [118]	-	-	75.4
Gao *et al*. [119]	-	-	74.9
Ours-TSSTN	**97.8**	80.2	86.3

Table 6.4: mAP(%) results on the Willow dataset.

Method	Bike-ride	Horse-ride	Run	Walk	Overall (Body-Motion)
Delaitre *et al*. [32]	82.43	69.60	44.53	54.18	62.7
Delaitre *et al*. [76]	90.39	75.03	59.73	57.64	70.7
Sharma *et al*. [109]	87.8	84.2	56.1	56.5	71.1
Sharma *et al*. [81]	91.0	87.6	55.0	59.2	73.2
Khan *et al*. [110]	87.2	77.2	63.7	60.6	72.2
Khan *et al*. [111]	93.8	87.9	67.2	63.3	78.05
Liang *et al*. [82]	98.17	92.72	46.16	58.88	74.0
Zhao *et al*. [80]	93.0	86.2	65.7	72.6	79.3
Khan *et al*. [42]	90.3	84.3	64.7	64.6	76.0
Ours-TSSTN	80.6	89.8	84.6	83.8	**84.6**

Table 6.5: Left: mAP (%) results on WIDER. - Right: mAP (%) results BU_{101} by categories.

Method	**mAP (%)**	
RCNN	80.0	
R*CNN	80.5	
DHC	81.3	
ResNet-SRN	86.2	
VeSPA	82.4	
Ours-TSSTN	Body-Motion	Non-Body-Motion
	90.3	71.7

Categories	**mAP (%)**
Human-Object	59.6
Body-Motion	93.8
Human-Human	68.9
Playing-Instrument	67.0
Sport	74.7

CHAPTER 7: LEARNING LATENT SPACE-TIME REPRESENTATION USING AN ENSEMBLE METHOD

In this chapter, we propose an ensemble method to learn a *Meta-Classifier*, comprising of *STCNN* and *ZTD*, the two heterogeneous base classifiers that we earlier proposed and studied in detail in chapters 3 and 4, respectively. While the first base classifier, *STCNN*, is highly dependent on training data for explicitly learning the latent space-time representations, *ZTD*, the second base classifier, is independent of training, relying solely on prototype groups of images that span the latent space-time domain for each action class. We expect that due to their heterogeneity, these two base classifiers would be complementary in terms of their per-class accuracy. Fig. 7.1 depicts an overview architecture of the proposed method. Moreover, in section 7.2, we show how creating tensor $\mathcal{Q}$ with different order of channels affect the classification accuracy. Extensive experiments on *UCF-STAR* and five other most popular datasets clearly demonstrate that ensembles of classifiers perform better than individual *STCNN* and *ZTD* classifiers. Also, we show how injecting prior knowledge into the domain mapping stage can improve the classification result.

7.1 Ensemble Learning

Our intuition for the choice of the two base classifiers is that their diverse nature makes them very much complementary, and hence extremely suitable for combining in an ensemble learning method in order to achieve a performance superior to both methods. An ensemble method is basically an array of classifiers, whose individual predictions are combined to form a new prediction model. There are mainly three approaches for building an ensemble method, i.e. bagging, boosting, and stacking. Since our classifiers are not applied sequentially, we cannot use a boosting approach. On the other hand, availability of training data suggest that stacking [145] would be the best option

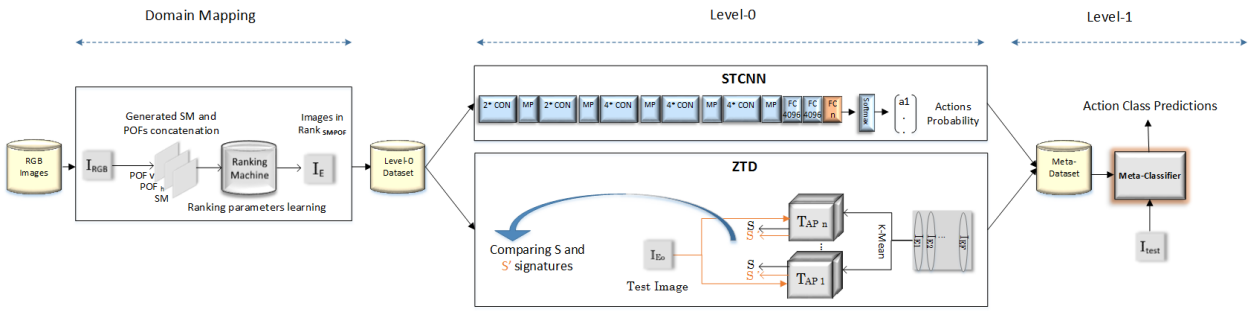

Figure 7.1: A schematic overview of the proposed single image action recognition framework. **Data Mapping**: RGB images are mapped onto the latent space-time domain $Rank_{\text{SM-POF}}$. **Level-0 (base-level) models**: *First base classifier* - CNN action classifier: Deep features from the activation of the fully connected layers of a STCNN model are used as input to a softmax layer to determine the final classification from the first base classifier. *Second base classifier* - Using sample images per action, 3-way prototype tensors are formed. Each prototype tensor spans the latent domain of the corresponding action class. Test image is classified as the label of the prototype tensor whose low-rank representation is closest to the test image. **Level-1 model**: *STCNN* and *ZTD* Level-0 models are combined to form a *meta-classifier* superior to the individual base classifiers.

in our case. In stacking, multiple classification or regression models are combined using a meta-classifier that is trained on the outputs of the base-level models as features to deliver the final classification results. The most common form of stacking involves as the first step, collecting the output of each model into a new set of data, called *Meta-dataset* or *Level-1 dataset*. For each instance in the original training set, this dataset represents every model's prediction of that instance's class, along with its true classification. The *Meta-dataset* is then treated as the training data for the meta-classifier to collectively solve the problem. The original data and the models constructed from them in the first step are referred to as the *Level-0* data and *Level-0* models, respectively. The set of cross-validated data and the second-stage learning algorithm are referred to as the *Level-1* data and the *Level-1* generalizer. Here, we use the output probabilities produced by our base classifiers as Level-0 models described to form the Level-1 data. Moreover, for the Level-1 generalizer, we use a linear least squares regression adapted for the classification task.

Given a data set $\mathcal{D} = \{(y_n, x_n), n = 1, ..., N\}$, where y_n is the class value and x_n is a vector representing the attribute values of the n_{th} instance, we randomly split the data into J almost equal parts $\mathcal{D}_1, .., \mathcal{D}_j$. Define $\mathcal{D}$ and $\mathcal{D}^{-j} = \mathcal{D} - \mathcal{D}_j$ to be the test and the training sets for for the j_{th} fold of a J-fold cross validation. Given k learning algorithms, which we call *Level-0* generalizers, invoke the k_{th} algorithm on the data in the training set $\mathcal{D}^{-j}$ to induce a model $\mathcal{M}_k^{-j}$ on x_n. At the end of the entire cross-validation process, the *meta*-dataset assembled from the outputs of the k models is:

$$\acute{\mathcal{D}} = (y_n, z_{1n}, ..., z_{kn}) \quad where \quad n = 1, ..., N \quad k = 1, ..., K \tag{7.1}$$

These are the *Level-1* data. We use a learning algorithm that we call the *Meta-Level* algorithm or *Level-1* generalizer to derive from these data a model $\acute{\mathcal{M}}$ for y as a function of $\{z_1, ..., z_K\}$. $\acute{\mathcal{M}}$ is the *Level-1* model. To complete the training process, the final level-0 models $\mathcal{M}_k$ are derived using all the data in $\mathcal{D}$. In the classification process we use the models $\mathcal{M}_k$, in conjunction with $\acute{\mathcal{M}}$. Given a new instance, models $\mathcal{M}_k$ produce vectors z_k. These vectors are input to the level-1 model $\acute{\mathcal{M}}$, whose output is the final classification result for that instance. Here, we consider a situation where the output from level-0 models is a set of class probabilities rather than a single class prediction. If model $\mathcal{M}_k^j$ is used to classify an instance in $\mathcal{D}_j$, then let $\mathcal{P}_{ki}(x)$ denote the probability of the i_{th} output class, and the vector

$$\mathcal{P}_{kn} = (\mathcal{P}_{k1}(x_n), ..., \mathcal{P}_{ki}(x_n), ..., \mathcal{P}_{kI}(x_n)) \tag{7.2}$$

give the model's class probabilities for the n_{th} instance, assuming that there are I classes. As the Level-1 data, assembled together the class probability vector form the k models, along with the

actual class:

$$\begin{aligned} \acute{\mathcal{D}}_{\circ} &= \{(y_n, \mathcal{P}_{1n}, ..., \mathcal{P}_{kn}, ..., \mathcal{P}_{Kn})\} \\ where \quad & n = 1, ..., N, k = 1, ..., K \end{aligned} \tag{7.3}$$

The Level-1 model derived from $\acute{\mathcal{D}}_{\circ}$ is denoted as $\acute{\mathcal{M}}_{\circ}$ to contrast with $\acute{\mathcal{M}}$.

Stacking with Multi-response Linear Regression (MLR) makes use of linear regression in order to perform classification [140]. MLR is an adaptation of a least-squares linear regression algorithm that Breiman [146] leveraged in the regression settings. Any classification problem with real-valued attributes can be transformed into a MLR problem. For instance, if the original classification problem has I classes, the problem is converted into I separate regression problems, where the problem for class c has instances with responses equal to one when they have class c and zero otherwise.

For our ensemble method, the input to MLR is level-1 data, and we need to consider the situation for the $\acute{\mathcal{M}}_{\circ}$ model, where the attributes are probabilities. The linear regression for class c is simply:

$$\mathcal{R}_c(x) = \sum_{k=1}^{K} \alpha_{kc} \mathcal{P}_{kc}(x) \tag{7.4}$$

Where we choose the linear regression coefficients $\{\alpha_{kc}\}$ to minimize

$$\sum_{d} \sum_{(y_n, x_n) \in \mathcal{D}_j} (y_n - \sum_{k} \alpha_{kc} \mathcal{P}_{kc}^{-j}(x_n))^2 \tag{7.5}$$

The non-negative-coefficient least-square algorithm described by [147] is employed here to derive the linear regression for each action class. Finally, to classify a new instance x, $\mathcal{R}_c(x)$ for all C

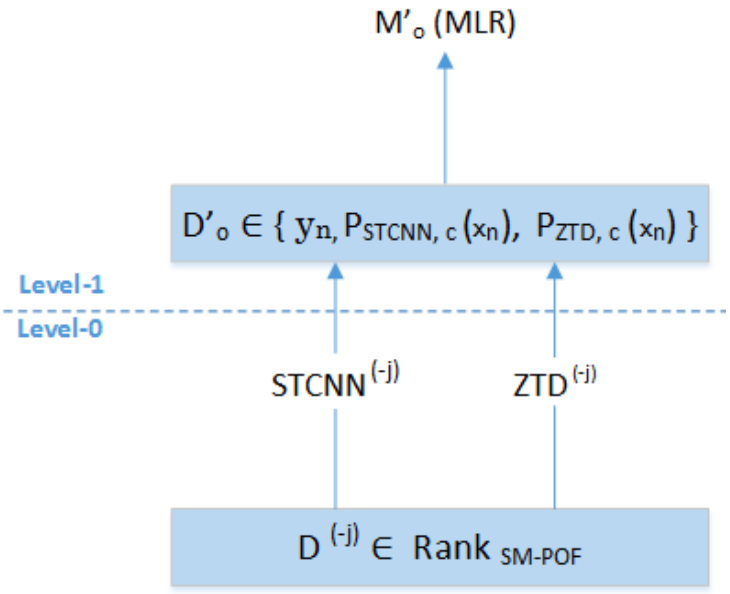

Figure 7.2: Illustration of the j-fold cross validation process in Level-0; STCNN and ZTD, the two level-0 classifiers, are employed over images in $Rank_{\text{SM-POF}}$, the latent space-time domain. The Level-1 dataset $\acute{\mathcal{D}}_\circ$ is used to produce the Level-1 model $\acute{\mathcal{M}}_\circ$. y_n is the class value for the n_{th} instance. $P_{STCNN,c}(x_n)$ and $P_{ZTD,c}(x_n)$ denote the probabilities of the instance x_n belonhgs to class c according to STCNN and ZTD models, respectively.

classes is computed and the instance x is assigned to the class c with the greatest value:

$$\mathcal{R}_c(x) > \mathcal{R}_{\acute{c}}(x) \quad for \quad all \quad c \neq \acute{c} \tag{7.6}$$

Fig. 7.2 illustrates the ensemble learning processes. Our empirical evaluation of all mentioned stages in our proposed framework will be extensively presented in the next section.

7.2 Evaluation and Experiments

As part of our experiments in this section, we constructed tensor $\mathcal{Q}$ (proposed in section 3.1) from the images in *UCFSI-101* and *UCF-STAR*, using three different orders of the space-time slices. As shown in Fig. 7.3, the classification accuracy over all action classes is not considerably affected by changing the order of the space-time slices in the raw tensor $\mathcal{Q}$. Although the overall accuracy is not a function of the order of space-time slices in $\mathcal{Q}$, per class accuracy does show variations, indicating that the order of the space-time slices before learning the latent feature space with Rank-SVM may be used as a prior to improve some class recognition accuracies. Our interpretation is that for actions that include more significant body motion temporal slices may play more important role than the spatial slice, and the prior choice of ordering would enforce a "preferred" prior expected ranking as input to Rank-SVM learning. In the next two sections, we discuss the idea of incorporating prior knowledge in the latent space-time domain learning process.

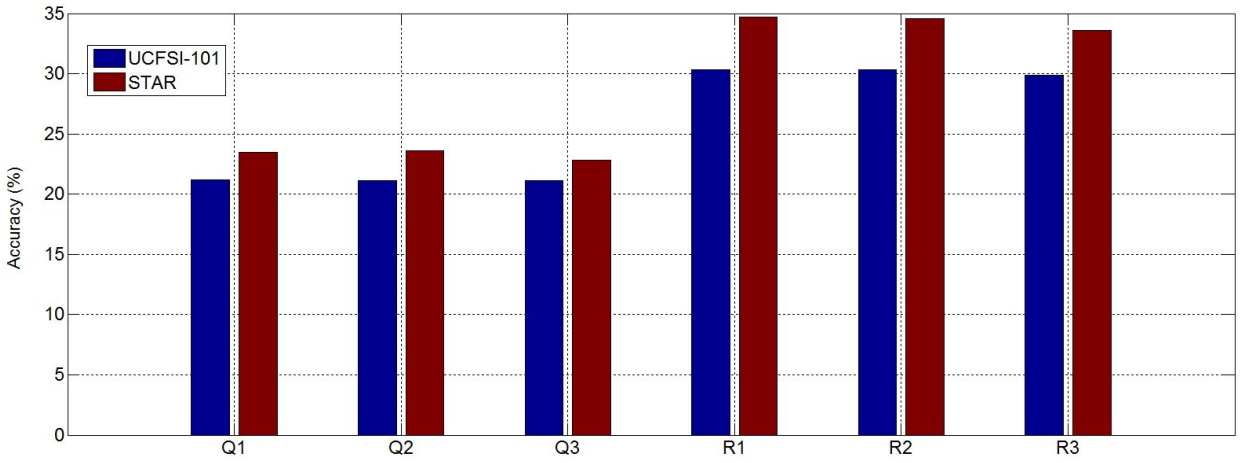

Figure 7.3: Action classification accuracy of STCNN, trained from scratch, as a function of the order of the slices in the tensor $\mathcal{Q}$. $\{$SM, POF$_h$, POF$_v\}$, $\{$SM, POF$_v$, POF$_h\}$ and $\{$POF$_h$, POF$_v$, SM$\}$ represent order of slices in $\mathcal{Q}1$, $\mathcal{Q}2$ and $\mathcal{Q}3$, respectively. $\mathcal{R}1$, $\mathcal{R}2$ and $\mathcal{R}3$ represent the mapping in the new domain after applying Rank-SVM to the tensors $\mathcal{Q}1$, $\mathcal{Q}2$ and $\mathcal{Q}3$, respectively.

7.2.1 Learning Latent Space-Time Domain Without Incorporating Prior Knowledge

Here, all images are treated equally regardless of which class they belong to. This implies that the tensor $\mathcal{Q}$ is constructed with the same order of space-time slices for all the images, regardless of action class. Therefore, the learning of the latent space-time domain through Rank-SVM does not include any prior knowledge on the order of importance of the time-space slices in $\mathcal{Q}$.

7.2.2 Learning Latent Space-Time Domain With Incorporating Prior Knowledge

In the context of learning the latent space-time feature space, prior knowledge may be viewed as the expected order of significance of spatial versus latent temporal information for a given action class. For instance, for a given class latent motions may be mostly horizontal or mostly vertical, while for others spatial relations may play a bigger role in discriminating the actions. Therefore the expected order of significance (if known as prior knowledge) can be enforced by ordering the space-time slices in $\mathcal{Q}$ differently.

While such global expected prior is enforced by a predefined ordering of the slices in $\mathcal{Q}$, the posterior pixelwise ranking function I_E is learned by applying Rank-SVM for each action class. We assumed that for body-motion categories, Predicted Optical Flows (POFs) were expected to be more significant, and hence were inserted as the first two slices of the tensor $\mathcal{Q}$, whereas for non-body-motion categories, Saliency Map (SM) was expected to be more significant, and was inserted as the first slice in the tensor $\mathcal{Q}$. Hereafter, we use $\mathcal{Q}_{\mathcal{B}}$ and $\mathcal{Q}_{\mathcal{NB}}$ notations for the body-motion and non-body-motion tensors, respectively. Similarly, I_{E_B} and $I_{E_{NB}}$ denote their corresponding learned ranking models, respectively.

In sections 7.3.2 and 7.4.2, we will demonstrate the benefit of incorporating prior knowledge in the proposed domain mapping approach in terms of improving accuracy and average precision in

classification.

7.3 Evaluation of STCNN Base Classifier

In chapter 3 we fully evaluated *STCNN*, a deep network fine-tuned over images in the $Rank_{\text{SM-POF}}$ latent domain, to perform action classification and compare it's performance against recent state-of-the-arts approaches. Here, We evaluate *STCNN* without and with incorporating prior knowledge into the network for classifying human actions in still images, in sections 7.3.1 and 7.3.2, respectively.

7.3.1 STCNN without Prior Knowledge

As discussed in section 7.2.1, we treated all images equally in the training phase regardless of which category they belong to, by generating tensors $\mathcal{Q}$ with the same order of slices. Next, I_E representations were generated for images in all datasets. We then removed the last fully-connected layer from the network, which was pre-trained on I_E representations of images in UCFSI-101 dataset and trained a linear softmax classifier for all other datasets. We stress that test and training images in all datasets including UCFSI-101 and *UCF-STAR*, are mutually exclusive. Extensive experiments on six benchmarks will be discussed further.

7.3.2 STCNN with Prior Knowledge

Here we focus on introducing prior knowledge into training data, to train *STCNN* for action classification, which however will not be available at test time. First, images in the training sets are divided into two categories; body-motion and non-body-motion, as described earlier in section

3.4.1 for all datasets. Second, as explained in section 7.2.2, tensors $\mathcal{Q}_{\mathcal{B}}$ and $\mathcal{Q}_{\mathcal{NB}}$ with different order of slices are created for images based on the category they belong to. Therefore, two independent networks are trained over I_{E_B} and $I_{E_{NB}}$ image representations to learn two independent models. At the test time, prior knowledge about action categories is not available. Therefore, given a test image, two I_{E_B} and $I_{E_{NB}}$ representations are generated. Finally, the two generated I_Es are separately fed into their corresponding learned model for action classification. We then use the class probabilities as a proxy for the degree of confidence in that classification. Hence, the test image is classified with the action class with the highest score.

As shown in tables 7.1-7.4, *STCNN* achieved promising results compared to state-of-the art approaches on all datasets. As mentioned in earlier chapters, considering that we focus on the human body salient pixels and their motion direction, we achieved the best performance on the body-motion group on all datasets. We also show in tables 7.1-7.4, incorporating *prior knowledge* into the domain mapping approach at the training phase improves the action classification performance in all experiments.

7.4 Evaluation of Base Classifier ZTD

As explained in chapter 4, we form groups of action prototypes by clustering I_Es, columns in matrix $\mathcal{R}$, based on the similarity between their latent space-time representations. Size of images are the same in all action prototype groups. Each group provides evidence for the distinctive compact latent spatiotemporal descriptions of images in that group. Relatively small groups are dropped as they correspond to limited numbers of images, illustrating infrequent action phases in an action class. An image, I_E is pruned out of its action prototype group if the distance of its corresponding entry in S is more than three scaled median absolute deviations away from the local median within a five-element window. Such images fail the condition of latent spatiotemporal

similarity with their action prototype group. The evaluation of *ZTD* without and with incorporating prior knowledge into the process of creating action prototype groups will be discussed in sections 7.4.1 and 7.4.2, respectively.

7.4.1 ZTD Without Prior Knowledge

As explained in section 7.2.1, all images are treated equally in order to create tensors $\mathcal{Q}$ followed by generating I_E representations, regardless of which action category they belong to. 325, 123, 25, 134 and 78 action prototype groups are created for UCFSI-101, *UCF-STAR*, Willow, Stanford-40 and WIDER datasets, respectively, as discussed in section 4.4. Every test image I_{E_o} has certain conditional probability distribution over all $\mathcal{T}_{ka_n}$, denoted by $P(\mathcal{T}_{ka_n}|I_{E_o})$. There is no action label available for I_{E_o}. I_{E_o} is plugged in the tensor $\mathcal{T}_{ka_n}$ at any location h and the vector S is perturbed at and around index h.

7.4.2 ZTD With Prior Knowledge

Here, prior knowledge is embedded in the process of creating action prototype groups. As explained in section 7.2.2, images in body-motion and non-body-motion categories are mapped onto I_{E_B} and $I_{E_{NB}}$, respectively. Action prototype groups are then created based on their corresponding I_E representations. Since there is no prior knowledge available at test time, given test image I_o, both $I_{E_{o_B}}$ and $I_{E_{o_{NB}}}$ representations are generated. Finally, $I_{E_{o_B}}$ and $I_{E_{o_{NB}}}$ are inserted in the body-motion and non-body-motion action prototype, respectively.

Tables 7.1-7.4 show *ZTD*'s performance on all benchmarks. Injecting prior knowledge into the action prototype groups improves the action classification accuracy by a considerable margin.

Table 7.1: Results (mAP) on Willow dataset.

Method	**Non-Body-Motion**	**Body-Motion**
Delaitre et al. [32]	55.6	62.7
Delaitre et al. [76]	55.4	70.7
Sharma et al. [109]	59.0	71.1
Sharma et al. [81]	60.1	73.2
Khan et al. [110]	62.4	72.2
Khan et al. [111]	64.1	78.05
Zhao et al. [80]	67.8	79.3
Khan et al. [42]	62.2	76.0
STCNN- without prior knowledge	64.5	**78.7**
STCNN- with prior knowledge	74.8	**80.6**
ZTD- without prior knowledge	75.3	**81.8**
ZTD- with prior knowledge	82.4	**84.9**

Table 7.2: Results (mAP) on Stanford-40.

Method	**Body-Motion**	**Non-Body-Motion**	**All**
Gkioxari et al. [112]	93.87	89.73	90.9
Khan et al. [111]	56.92	51.51	53
Khan et al. [42]	53.51	51.28	51.9
Yan et al. [113]	92.26	87.07	88.5
Zhao et al. [114]	-	-	83.4
Zhao et al. [80]	-	-	54.5
Zhao et al. [115]	-	-	80.6
Zhou et al. [116]	-	-	55.3
Sharma et al. [117]	-	-	72.3
Khan et al. [118]	-	-	75.4
Gao et al. [119]	-	-	74.9
STCNN without prior knowledge	**94.3**	73.1	81.76
STCNN with prior knowledge	**95.4**	84.7	88.5
ZTD without prior knowledge	**95.5**	78.2	82.9
ZTD with prior knowledge	**96.3**	87.6	87.8

7.5 Evaluation of The Ensemble Method

At the meta-level, we evaluate the performance of three different schemes for combining classifiers. As described in section 7.1, stacking with MLR uses linear regression to perform classification. For a classification problem with C classes, C regression problems are formulated.

Table 7.3: Results (mAP) on WIDER.

Method	**mAP**(%)	
RCNN	80.0	
R*CNN	80.5	
DHC	81.3	
ResNet-SRN	86.2	
VeSPA	82.4	
STCNN without prior knowledge	Body-Motion	Non-Body-Motion
	86.8	59.5
STCNN with prior knowledge	Body-Motion	Non-Body-Motion
	88.4	68.9
ZTD without prior knowledge	Body-Motion	Non-Body-Motion
	88.7	72.7
ZTD with prior knowledge	Body-Motion	Non-Body-Motion
	89.3	80.9

Table 7.4: Results (mAP) on BU_{101} dataset.

mAP (%)					
Category	Human-Object	Body-Motion	Human-Human	Playing-Instrument	Sport
STCNN without prior knowledge	61.1	84.4	58.7	71.3	74.8
STCNN with prior knowledge	65.0	86.1	60.9	71.6	77.9
ZTD without prior knowledge	58.9	79.9	59.4	69.4	73.6
ZTD with prior knowledge	64.8	87.6	59.1	70.1	74.2

Given a new example x to classify, $\mathcal{R}_c(x)$ is calculated for all $c \in C$, and the class c is predicted with the maximum $\mathcal{R}_c(x)$. Moreover, we consider two baselines to which the *stacking* with *MLR* performance should be compared. *Un-Weighted Averaging (UWA)*: The predictions of each base classifier become columns in a matrix where rows are instances' probabilities and the entry at row i, column j is the probability of instance x belonging to the class i as predicted by classifier j. We apply the mean across rows to produce an aggregate prediction for each instance; *Select Best Base classifier (SBB)*: This general approach tries to identify the best base classifier among the set of base learners for a specified input, and the output of the ensemble is the output of the selected best classifier.

In all the experiments presented here, classification errors are estimated using 10-fold cross validation. Cross validation is repeated ten times using different random generator seeds resulting in ten different sets of folds. The same folds are used in all the experiments. The classification error of a classification algorithm k for a given dataset as estimated by averaging over the ten runs of ten-fold cross validation is denoted E_k.

First, we compare error rate of $\acute{\mathcal{M}}_\circ$, derived from (7.4), to the baselines algorithms which require neither cross-validation nor Level-1 algorithm. In order to see whether the relative performances of level-0 generalizers have any effect on these methods, the number of standard errors (SE) between the error rates of the two level-0 generalizers is given. Since SBB almost always selects the best performing level-0 generalizer, small values of SE indicate that the level-0 generalizers perform comparably to one another, and vice versa. Table 7.5 represents the average error rates of baselines and MRL. Because we are using heterogeneous classifiers that may have uncalibrated outputs, the mean combines predictions made with different scales or notions of probability. This explains its poor performance compared to the best base classifier in a heterogeneous ensemble and emphasizes the need for ensemble selection or weighting via stacking to take full advantage of the ensemble.

Table 7.5: Average error rates of UWA, SBB and the MLR (model $\acute{\mathcal{M}}_\circ$), along with the number of standard error between Level-0 classifiers.

Dataset	**SE**	**UWA**	**SBB**	**MLR**
UCFSI-101	12.6	29.1	22.8	**17.4**
UCF-STAR	1.6	20.1	19.3	**15.7**
Willow	6.4	20.8	17.6	**12.9**
Stanford-40	6.6	13.4	10.1	**7.6**
WIDER	33.6	35.7	**18.9**	19.2

For pair-wise comparison of classification algorithms, we calculate the relative improvement. In order to evaluate the accuracy improvement achieved in a given domain by using classifier k_1 as

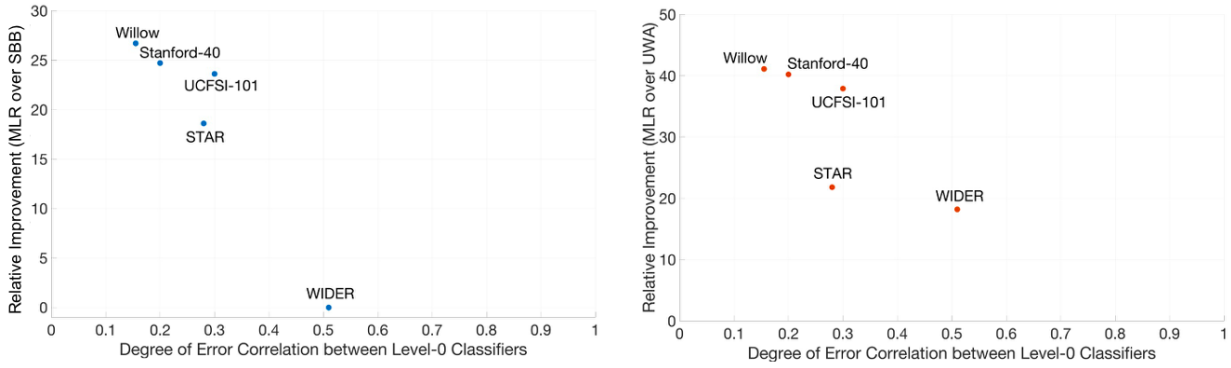

Figure 7.4: The relationship between error correlation and relative improvement of MLR over SBB and UWA.

compared to using classifier k_2, we calculate the relative improvement: $1 - E_{k_1}/E_{k_2}$. In table 7.6, we compare the performance of MLR to other combining approaches.

Table 7.6: The relative performance of ensembles with different combining method. The X-Y gives the relative improvement of X over Y in %

Datasets	**Level-1**		
	MLR-UWA	MLR-SBB	SBB-UWA
UCFSI-101	40.2	23.6	36.7
UCF-STAR	21.8	18.6	24.5
Willow	37.9	26.7	18.2
Stanford-40	41.0	24.7	11.1
WIDER	18.2	-0.015	29.8

We also show the improvement of the performance of MLR over UWA and BBS is related to the diversity of the Level-0 classifiers. We use the measure of error correlation i.e. higher diversity means lower error correlation; proposed by [148]. For two classifiers k_i and k_j, this measure $\phi(k_i, k_j)$ is defined as the conditional probability that both classifiers make the same error, given

Table 7.7: Performance (mAP) on *UCF-STAR* dataset against state of the arts.

Method	**mAP**
Object Bank [143]	26.7
LLC [144]	31.5
R*CNN [135]	71.3
STCNN (First Level-0 Classifier)	**79.1**
ZTD (Second Level-0 Classifier)	**80.6**
MLR (Ensemble of Level-0 Classifiers)	**86.3**

Table 7.8: Comparing MLR results (mAP) against level-0 algorithms

Dataset	Body-Motion			Non-Body-Motion		
	STCNN	ZTD	MLR	STCNN	ZTD	MLR
UCFSI-101	89.1	91.0	93.2	63.3	65.8	66.5
Willow	78.7	81.8	85.0	64.5	75.3	75.5
Stanford-40	94.3	95.5	96.3	73.1	78.2	83.7
WIDER	86.8	88.7	91.2	59.5	67.8	67.6

that one of them makes an error:

$$p(k_i(x) = k_j(x) | k_i(x) \neq k(x) \vee k_j(x) \neq k(x)) \tag{7.7}$$

where $k_i(x)$ and $k_j(x)$ are the prediction of the classifiers k_i and k_j for a given example x and $k(x)$ is the true class of x. The error correlation for a set of multiple classifiers K is defined as the average of the pairwise error correlation:

$$\phi(K) = \frac{1}{|K|(|K|-1)} \sum_{k_i \in K} \sum_{k_i \neq k_j} \phi(k_i, k_j) \tag{7.8}$$

Fig. 7.4 depicts the relative improvement as a function of the degree of error correlation of MLR over UWA and BBS. It is clear that relative improvement increases as error correlation decreases

(the lower the error correlation, the higher the diversity). This indicates that MLR uses the diversity of the Level-0 classifiers better than the competing combining methods.

Tables 7.7 and 7.8 shows that ensembles of classifiers induced by stacking with MLR, perform better than individual *STCNN* and *ZTD* classifiers, otherwise the extra work of learning a meta-level classifier doesn't not seem justified. Thus the extra computation for cross-validation and level-1 learning seems to have paid off.

CHAPTER 8: CONCLUSION

One of the main goals in our work was to design algorithms that leverage the insights from both multi-linear and modern deep learning approaches. Our system has potential to become significant tool for problems of limited initial information. The main motivation behind this dissertation was to address the problem of absence of temporal information in still images that poses a huge challenge in single image action recognition. In recent years, action recognition in still images has gained increasing attention due to its challenging nature and its importance in applications, such as video storage and retrieval [15, 2, 16], automated video surveillance [3, 4, 5], modeling surveillance camera networks [6, 7, 8], image context analysis and annotation [9, 10], humanmachine interface [11, 12] and identity recognition [13]. We have presented multiple significant contributions to this field by modeling the *latent temporal information* in still images, and using it as prior knowledge toward action recognition in still images.

Chapter 3 presented a novel domain mapping technique to capture spatiotemporal evolution of pixels in order to obtain a unique compact latent space-time representation for a single image. This mapping is based on the Rank-SVM algorithm, projecting images onto a new domain, named *Ranked Saliency Map and Predicted Optical Flow*, or $Rank_{\text{SM-POF}}$ for short. We proposed to use the predicted optical flow in a static image as a means of compensating for the missing temporal information, while using the saliency map to represent the spatial information about the location and the shape of the predicted significant regions of the image. Hence, the saliency map and the predicted optical flow are employed to covert the raw still image to a novel image representations that capture both spatial salient part of the actor as well as the future movement patterns of the actor by learning the functional parameters of the linear ranking functions. Action classification is then treated as a transfer learning problem, where *STCNN*, a new Spatial-Temporal Convolutional Neural Network model is trained by fine-tuning a CNN model, pre-trained on appearance-based

classification only. *STCNN* is trained to classify human actions based on both spatial and temporal features; i.e. using the images in the $Rank_{SM-POF}$ as input. In other terms, we did supervised domain adaptation via fine-tuning a pre-trained network on the new domain. We carried on a comprehensive analysis on both domain mapping and transfer learning components presented in this chapter. Results on multiple benchmarks demonstrated that appearance and motion are complementary sources of information, hence using both leads to significant performance improvements in single image action recognition. This work is published in IEEE Winter Conference on Applications of Computer Vision.

Chapter 4 introduced a novel unsupervised Zero-shot approach based on low-rank *Tensor Decomposition*, named *ZTD*. We introduced a unique solution to form action prototype tensors in a way that each tensor encodes useful information regarding latent spatiotemporal patterns of images. We used the domain mapping approach introduced in chapter 3 to model the latent space-time domain of each action class, using a group of images per action class. We assumed that, although a single image may not uniquely characterize its latent space-time domain, a group of sample images for an action would form an action prototype that would span the latent domain of the corresponding action class. The image representations in one action prototype group should have sufficient visual similarity so that one group could be used as basis for formation of spatiotemporal signature of an action. Tensor decomposition is then used to find out how individual elements of one group relate to the overall group so that the classifier may determine if some entity belongs to the group or not. Applying tensor decomposition on a tensor made up of images having similar spatiotemporal behavior belonging to one action prototype group, generated an action signature, a vector ideally with little variation across its entries. Hence, inserting a test image into the action prototype tensors disturbed entries in the signature vector. The amount of disturbance will be proportional to the dissimilarity between the test image and other members of that action prototype group. Action classification for a test image is then treated as recognizing the prototype group whose low-rank

representation is closest to the test image, i.e. by measuring to what extent the test image would perturb such low-rank representation for each class. Unlike the method presented in chapter 4, *ZTD* is independent of training, relying solely on prototype groups of images that span the latent space-time domain for each action class. This work can therefore be summarized as a training independent classifier yet improving action classification performance compared to state of the arts. This work is published in IEEE International Conference on Image Processing.

Chapter 5 introduced a new image representation, named *temporal image*, by learning motion from video frames and then transferring it to still images. Therefore, a temporal image served to model the missing temporal information by representing motions of human body pixels in a predefined time window, hallucinating the human body motion. In this chapter, we also proposed *TICNN*, a hierarchical deep learning framework for still image action recognition using temporal image prediction. *TICNN* was the first attempt to predict a temporal image, dynamic patterns of a still image, when the video data is not available. We designed a CNN that used the capabilities of pixel-wise image segmentation to learn a model that predicts a temporal image given a still image. A predicted temporal image represents a compact representation of hypothetical sequence of images of a non-existing video. These temporal patterns are produced as an auxiliary form of information for still image action recognition showing that *TICNN*'s results are superior. This work is publishes in IEEE International Conference on Image Processing and received the best paper award.

chapter 6 introduced *UCF-STAR*, the largest annotated still image dataset in the literature for action recognition, having over $1M$ images, collected from the *wild*, across 50 different human *body-motion* action categories, annotated with multi-modal set of metadata. *UCF-STAR* is created, to advance the current still image-based action recognition research, and to promote future research opportunities through its additional rich and structured metadata. In addition, we proposed *TSSTN*, a two stream spatiotemporal network that outperforms the current state of the art on *UCF-STAR* to serve as a baseline. *UCF-STAR* highlights the need for developing new action recognition ap-

proaches based on predicting temporal information in still images. This work is publishes in AAAI Conference on Artificial Intelligence.

Chapter 7 proposed an ensemble method to learn a *Meta-Classifier* by comprising of *STCNN* and *ZTD*, the two heterogeneous base classifiers. While the first base classifier, *STCNN*, is highly dependent on training data for explicitly learning the latent space-time representations, *ZTD*, the second base classifier, is independent of training. Extensive experiments demonstrated that due to their heterogeneity, the two base classifiers are complementary in terms of their per-class accuracy. Hence, the ensembles of classifiers outperforms the individual *STCNN* and *ZTD* classifiers. Moreover, we demonstrated that how incorporating prior knowledge in the latent space-time domain learning process domain can improve the classification performance.

This work can therefore be summarized as injecting the latent temporal information into still images, hence treating temporal information as context, while related efforts mainly consider only spatial information as context. Extensive experiments over nine datasets proved that employing the latent temporal information as prior knowledge in still images improves action recognition performance, especially in human body-motion action categories. Our system is more challenged when exposed to images portraying stationary actions. Differentiating between stationary and body-motion actions prior to action recognition, and further making use of this information in the classification process, could be an interesting future work. Finally, an extension of individual action recognition to further perform group action recognition is also an area with tremendous amount of attention in the literature.

APPENDIX A: IEEE COPYRIGHT INFORMATION

Still Image Action Recognition by Predicting Spatial-Temporal Pixel Evolution

Conference Proceedings:
2019 IEEE Winter Conference on Applications of Computer Vision (WACV)

Author: [::Marjaneh::] [::Safaei::]; Hassan Foroosh

Publisher: IEEE

Date: 7-11 Jan. 2019

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

APPENDIX B: IEEE COPYRIGHT INFORMATION

A Zero-Shot Architecture for Action Recognition in Still Images

Conference Proceedings: 2018 25th IEEE International Conference on Image Processing (ICIP)

Author: [::Marjaneh::] [::Safaei::]; Hassan Foroosh

Publisher: IEEE

Date: 7-10 Oct. 2018

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

APPENDIX C: IEEE COPYRIGHT INFORMATION

TICNN: A Hierarchical Deep Learning Framework for Still Image Action Recognition Using Temporal Image Prediction

Conference Proceedings: 2018 25th IEEE International Conference on Image Processing (ICIP)

Author: Marjaneh Safaei; Pooyan Balouchian; Hassan Foroosh

Publisher: IEEE

Date: 7-10 Oct. 2018

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

APPENDIX D: AAAI COPYRIGHT INFORMATION

Association for the Advancement of Artificial Intelligence

2275 East Bayshore Road, Suite 160
Palo Alto, California 94303 USA

AAAI COPYRIGHT FORM

Title of Article/Paper: UCF-STAR: A Large-Scale Still Image Dataset for Understanding Human Actions

Publication in Which Article/Paper Is to Appear: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAA

Author's Name(s): Marjaneh Safaei, Pooyan Balouchian, Hassan Foroosh

Please type or print your name(s) as you wish it (them) to appear in print

PART A – COPYRIGHT TRANSFER FORM

The undersigned, desiring to publish the above article/paper in a publication of the Association for the Advancement of Artificial Intelligence, (AAAI), hereby transfer their copyrights in the above article/paper to the Association for the Advancement of Artificial Intelligence (AAAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications.

This grant will include, without limitation, the entire copyright in the article/paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights current exist or hereafter come into effect, and also the exclusive right to create electronic versions of the article/paper, to the extent that such right is not subsumed under copyright.

The undersigned warrants that they are the sole author and owner of the copyright in the above article/paper, except for those portions shown to be in quotations; that the article/paper is original throughout; and that the undersigned right to make the grants set forth above is complete and unencumbered.

If anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the undersigned will hold harmless and indemnify AAAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense AAAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to AAAI in the article/paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorneys' fees incurred therein.

Returned Rights

In return for these rights, AAAI hereby grants to the above author(s), and the employer(s) for whom the work was performed, royalty-free permission to:

1. Retain all proprietary rights other than copyright (such as patent rights).
2. Personal reuse of all or portions of the above article/paper in other works of their own authorship. This does not include granting third-party requests for reprinting, republishing, or other types of reuse. AAAI must handle all such third-party requests.
3. Reproduce, or have reproduced, the above article/paper for the author's personal use, or for company use provided that AAAI copyright and the source are indicated, and that the copies are not used in a way that implies AAAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the article/paper in electronic or digital form on any computer network, except by the author or the author's employer, and then only on the author's or the employer's own web page or ftp site. Such web page or ftp site, in addition to the aforementioned requirements of this Paragraph, shall not post other AAAI copyrighted materials not of the author's or the employer's creation (including tables of contents with links to other papers) without AAAI's written permission.
4. Make limited distribution of all or portions of the above article/paper prior to publication.
5. In the case of work performed under a U.S. Government contract or grant, AAAI recognized that the U.S. Government has royalty-free permission to reproduce all or portions of the above Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract or grant so requires.

In the event the above article/paper is not accepted and published by AAAI, or is withdrawn by the author(s) before acceptance by AAAI, this agreement becomes null and void.

(1) _____ 11/19/2019

Author/Authorized Agent for Joint Author's Signature — Date

_____ 11/21/2019

Employer for whom work was performed — Title (if not author)

(For jointly authored Works, all joint authors should sign unless one of the authors has been duly authorized to act as agent for the others.)

LIST OF REFERENCES

[1] M. Hosseinzadeh and Y. Wang, "Composed query image retrieval using locally bounded features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] C. Sun, I. Junejo, and H. Foroosh, "Motion retrieval using low-rank subspace decomposition of motion volume," *Computer Graphics Forum*, vol. 30, no. 7, pp. 1953–1962, 2011.

[3] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 1–43, 2011.

[4] I. Junejo and H. Foroosh, "Euclidean path modeling for video surveillance," *Image and Vision Computing*, vol. 26, no. 4, pp. 512–528, 2008.

[5] M. Ziaeefard and R. Bergevin, "Semantic human activity recognition: a literature review," *Pattern Recognition*, vol. 48, no. 8, pp. 2329–2345, 2015.

[6] X. Cao, J. Xiao, and H. Foroosh, "Self-calibration from turn table sequence in presence of zoom and focus," *Computer Vision and Image Understanding*, vol. 102, no. 3, pp. 227–237, 2006.

[7] I. Junejo and H. Foroosh, "Robust auto-calibration from pedestrians," in *Proceedings of IEEE International Conference on Advanced Video and Signal-based Surveillance*, 2006, pp. 92–97.

[8] I. Junejo, X. Cao, and H. Foroosh, "Autoconfiguration of a dynamic non-overlapping camera network," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 37, no. 4, pp. 803–816, 2007.

[9] A. Tariq and H. Foroosh, "Feature-independent context estimation for automatic image annotation," in *Proceedings of CVPR*, 2015.

[10] ——, "A context-driven extractive framework for generating realistic image descriptions," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 619–632, 2017.

[11] G. T. Papadopoulos, A. Axenopoulos, and P. Daras, "Real-time skeleton-tracking-based human action recognition using kinect data," in *International Conference on Multimedia Modeling*. Springer, 2014, pp. 473–483.

[12] L. L. Presti and M. La Cascia, "3d skeleton-based human action classification: A survey," *Pattern Recognition*, vol. 53, pp. 130–147, 2016.

[13] S. N. Paul and Y. J. Singh, "Survey on video analysis of human walking motion," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 3, pp. 99–122, 2014.

[14] C. Sun, M. Tappen, and H. Foroosh, "Feature-independent action spotting without human localization, segmentation or frame-wise tracking," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] J. C. Van Gemert, M. Jain, E. Gati, C. G. Snoek *et al.*, "Apt: Action localization proposals from dense trajectories." in *BMVC*, vol. 2, 2015, p. 4.

[16] H. Zhu, R. Vial, and S. Lu, "Tornado: A spatio-temporal convolutional regression network for video action proposal," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5813–5821.

[17] Y. Shen and H. Foroosh, "Methods for recognizing pose and action of articulated objects with collection of planes in motion," 2014, uS Patent 8,755,569.

[18] C. Sun, I. Junejo, and H. Foroosh, "Action recognition using rank-1 approximation of joint self-similarity volume," in *Proc. International Conference on Computer Vision (ICCV)*, 2012, pp. 1007–1012.

[19] C. Sun and H. Foroosh, "Should we discard sparse or incomplete videos?" in *Proc. of IEEE International Conference on Image Processing (ICIP)*, 2014.

[20] S. Cho and H. Foroosh, "A temporal sequence learning for action recognition and prediction," in *Proc. of IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[21] C. Sun, I. Junejo, and H. Foroosh, "Action recognition using rank-1 approximation of joint self-similarity volume," in *Proc. of IEEE Int. Conf. on Computer Vision*, 2011.

[22] N. Ashraf, C. Sun, and H. Foroosh, "View invariant action recognition using projective depth," *CVIU*, pp. 41–52, 2014.

[23] C. Sun, M. Tappen, and H. Foroosh, "Feature-independent action spotting without human localization, segmentation or frame-wise tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2689–2696.

[24] C. Sun, I. Junejo, M. Tappen, and H. Foroosh, "Exploring sparseness and self-similarity for action recognition," *IEEE Trans.s on image processing*, pp. 2488–2501, 2015.

[25] M. Di Luca and D. Rhodes, "Optimal perceived timing: Integrating sensory information with dynamically updated expectations," *Scientific reports*, vol. 6, p. 28563, 2016.

[26] M. Safaei and H. Foroosh, "Still image action recognition by predicting spatial-temporal pixel evolution," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 111–120.

[27] ——, "A zero-shot architecture for action recognition in still images," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 460–464.

[28] M. Safaei, P. Balouchian, and H. Foroosh, "Ticnn: A hierarchical deep learning framework for still image action recognition using temporal image prediction," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2018.

[29] S. Ma, S. A. Bargal, J. Zhang, L. Sigal, and S. Sclaroff, "Do less and achieve more: Training cnns for action recognition utilizing action images from the web," *Pattern Recognition*, vol. 68, pp. 334–345, 2017.

[30] M. Safaei, P. Balouchian, and H. Foroosh, "Ucf-star: A large scale still image dataset for action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[31] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *ICCV*. IEEE, 2011.

[32] V. Delaitre, I. Laptev, and J. Sivic, "Recognizing human actions in still images: a study of bag-of-features and part-based representations," in *BMVC 2010*, 2010.

[33] Y. Xiong, K. Zhu, D. Lin, and X. Tang, "Recognize complex events from static images by fusing deep channels." IEEE, 2015.

[34] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[35] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176.

[36] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[37] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[38] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *European conference on computer vision*. Springer, 2008, pp. 650–663.

[39] Y. Zhang, L. Cheng, J. Wu, J. Cai, M. N. Do, and J. Lu, "Action recognition in still images with minimum annotation efforts," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5479–5490, 2016.

[40] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Computer Vision and Pattern Recognition*, 2014, pp. 1717–1724.

[41] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "R-cnns for pose estimation and action detection," *arXiv preprint arXiv:1406.5212*, 2014.

[42] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, A. M. Lopez, and M. Felsberg, "Coloring action recognition in still images," *International Journal of Computer Vision (IJCV)*, vol. 105, no. 3, pp. 205–221, 2013.

[43] G. Guo and A. Lai, "A survey on still image based human action recognition," *Pattern Recognition*, vol. 47, no. 10, pp. 3343–3361, 2014.

[44] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[45] L.-J. Li and F.-F. Li, "What, where and who? classifying events by scene and object recognition." in *Iccv*, vol. 2, no. 5, 2007, p. 6.

[46] N. Shapovalova, W. Gong, M. Pedersoli, F. X. Roca, and J. Gonzalez, "On importance of interactions and context in human action recognition," in *Iberian conference on pattern recognition and image analysis*. Springer, 2011, pp. 58–66.

[47] B. Yao and L. Fei-Fei, "Grouplet: A structured image representation for recognizing human and object interactions," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 9–16.

[48] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, CVPR*, vol. 1, 2005, pp. 886–893.

[49] C. Desai and D. Ramanan, "Detecting actions, poses, and objects with relational phraselets," in *European Conference on Computer Vision*. Springer, 2012, pp. 158–172.

[50] C. Thurau and V. Hlavác, "Pose primitive based human action recognition in videos or still images," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[51] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: Using spatial and functional compatibility for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1775–1789, 2009.

[52] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for static human-object interactions," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE, 2010, pp. 9–16.

[53] S. Belongie, G. Mori, and J. Malik, "Matching with shape contexts," in *Statistics and Analysis of Shapes*. Springer, 2006, pp. 81–105.

[54] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori, "Unsupervised discovery of action classes," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1654–1661.

[55] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*, 2010, pp. 17–24.

[56] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[57] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 601–614, 2011.

[58] P. Li and J. Ma, "What is happening in a still picture?" in *The First Asian Conference on Pattern Recognition*. IEEE, 2011, pp. 32–36.

[59] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision (IJCV)*, vol. 88, no. 2, pp. 303–338, 2010.

[60] P. Li, J. Ma, and S. Gao, "Actions in still web images: visualization, detection and retrieval," in *International Conference on Web-Age Information Management*. Springer, 2011, pp. 302–313.

[61] N. Ikizler, R. G. Cinbis, S. Pehlivan, and P. Duygulu, "Recognizing actions from still images," in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[62] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[63] L. Bourdev, S. Maji, and J. Malik, "Describing people: A poselet-based approach to attribute classification," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1543–1550.

[64] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 1365–1372.

[65] S. Maji, L. Bourdev, and J. Malik, "Action recognition from a distributed representation of pose and appearance," in *Computer Vision and Pattern Recognition, CVPR, 2011*. IEEE, 2011, pp. 3177–3184.

[66] Y. Zheng, Y.-J. Zhang, X. Li, and B.-D. Liu, "Action recognition in still images using a combination of human pose and context information," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 785–788.

[67] W. Yang, Y. Wang, and G. Mori, "Recognizing human actions from still images with latent poses," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2030–2037.

[68] K. Raja, I. Laptev, P. Pérez, and L. Oisel, "Joint pose estimation and action recognition in image graphs," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 25–28.

[69] B. Yao and L. Fei-Fei, "Action recognition with exemplar based 2.5 d graph matching," in *European Conference on Computer Vision*. Springer, 2012, pp. 173–186.

[70] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Computer Vision and Pattern Recognition, 2010*, 2010, pp. 73–80.

[71] F. Sener, C. Bas, and N. Ikizler-Cinbis, "On recognizing actions in still images via multiple features," in *European Conference on Computer Vision*. Springer, 2012, pp. 263–272.

[72] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1931–1947, 2006.

[73] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[74] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[75] D. T. Le, R. Bernardi, and J. Uijlings, "Exploiting language models to recognize unseen actions," in *ACM conference on International conference on multimedia retrieval*, 2013, pp. 231–238.

[76] V. Delaitre, J. Sivic, and I. Laptev, "Learning person-object interactions for action recognition in still images," in *Advances in neural information processing systems*, 2011, pp. 1503–1511.

[77] B. Yao, A. Khosla, and L. Fei-Fei, "Classifying actions and measuring action similarity by modeling the mutual context of objects and human poses," *a) A*, vol. 1, no. D2, p. D3, 2011.

[78] B. Yao and L. Fei-Fei, "Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1691–1703, 2012.

[79] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 601–614, 2012.

[80] Z. Zhao, H. Ma, and X. Chen, "Generalized symmetric pair model for action classification in still images," *Pattern Recognition*, vol. 64, pp. 347–360, 2017.

[81] G. Sharma, F. Jurie, and C. Schmid, "Expanded parts model for human attribute and action recognition in still images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 652–659.

[82] Z. Liang, X. Wang, R. Huang, and L. Lin, "An expressive deep model for human action parsing from a single image," in *Multimedia and Expo (ICME), 2014*. IEEE, 2014, pp. 1–6.

[83] M. Hoai12, "Regularized max pooling for image categorization," 2014.

[84] S. Cho, M. H. Maqbool, F. Liu, and H. Foroosh, "Self-attention network for skeleton-based human action recognition," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[85] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proceedings of CVPR*, 2015.

[86] M. Wang, B. Liu, and H. Foroosh, "Design of efficient convolutional layers using single intra-channel convolution, topological subdivisioning and spatial bottleneck," *arXiv preprint*, vol. arXiv:1608.04337, 2016.

[87] ——, "Factorized convolutional neural networks," in *Proceedings of ICCV*, 2017, pp. 545–553.

[88] T. G. Dietterich, "Machine-learning research," *AI magazine*, vol. 18, no. 4, pp. 97–97, 1997.

[89] M. Gams, M. Bohanec, and B. Cestnik, "A schema for using multiple knowledge," in *Proceedings of the workshop on Computational learning theory and natural learning systems (vol. 2): intersections between theory and experiment: intersections between theory and experiment*. MIT Press, 1994, pp. 157–170.

[90] L. Todorovski and S. Džeroski, "Combining multiple models with meta decision trees," in *European conference on principles of data mining and knowledge discovery*. Springer, 2000, pp. 54–64.

[91] C. J. Merz, "Using correspondence analysis to combine classifiers," *Machine Learning*, vol. 36, no. 1-2, pp. 33–58, 1999.

[92] I. Laptev, B. Caputo *et al.*, "Recognizing human actions: a local svm approach," in *null*. IEEE, 2004, pp. 32–36.

[93] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1395–1402.

[94] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[95] H. Jhuang, H. Garrote, E. Poggio, T. Serre, and T. Hmdb, "A large video database for human motion recognition," in *Proc. of IEEE International Conference on Computer Vision*, 2011.

[96] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[97] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[98] N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff, "Learning actions from the web," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 995–1002.

[99] J. Walker, A. Gupta, and M. Hebert, "Dense optical flow prediction from a static image," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2443–2451.

[100] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *Journal of vision*, vol. 9, no. 12, pp. 15–15, 2009.

[101] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[102] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[103] H. Yu and S. Kim, "Svm tutorialclassification, regression and ranking," in *Handbook of Natural computing*. Springer, 2012, pp. 479–506.

[104] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, "Rank pooling for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 773–787, 2017.

[105] Y. Xiong, K. Zhu, D. Lin, and X. Tang, "Recognize complex events from static images by fusing deep channels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1600–1609.

[106] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[107] ——, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[108] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[109] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *Computer Vision and Pattern Recognition, 2012*. IEEE, 2012, pp. 3506–3513.

[110] F. S. Khan, J. Van De Weijer, A. D. Bagdanov, and M. Felsberg, "Scale coding bag-of-words for action recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 1514–1519.

[111] F. S. Khan, J. van de Weijer, R. M. Anwer, M. Felsberg, and C. Gatta, "Semantic pyramids for gender and action recognition," *IEEE transactions on image processing*, vol. 23, no. 8, pp. 3633–3645, 2014.

[112] G. Gkioxari, R. B. Girshick, and J. Malik, "Contextual action recognition with r*cnn," *CoRR*, 2015.

[113] S. Yan, J. S. Smith, and B. Zhang, "Action recognition from still images based on deep vlad spatial pyramids," *Signal Processing: Image Communication*, 2017.

[114] Z. Zhao, H. Ma, and S. You, "Single image action recognition using semantic body part actions," *CoRR*. [Online]. Available: http://arxiv.org/abs/1612.04520

[115] Z. Zhao, H. Ma, and X. Chen, "Semantic parts based top-down pyramid for action recognition," *Pattern Recognition Letters*, 2016.

[116] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014.

[117] G. Sharma, F. Jurie, and C. Schmid, "Expanded parts model for semantic description of humans in still images," *TPAMI*, 2017.

[118] F. S. e. a. Khan, "Recognizing actions through action-specific person detection," *IEEE transactions on image processing*, 2015.

[119] R. Gao, B. Xiong, and K. Grauman, "Im2flow: Motion hallucination from static images for action recognition," in *Proc. IEEE CVPR*, 2018.

[120] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs, "Nbclust: An r package for determining the relevant number of clusters in a data set," *Journal of Statistical Software*, vol. 61, pp. 1–36, 2014.

[121] Y.-D. Kim and S. Choi, "Nonnegative tucker decomposition," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[122] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[123] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[124] B. Chen, Z. Li, and S. Zhang, "On tensor tucker decomposition: the case for an adjustable core size," *Technical Report, University of Minnesota*, 2013.

[125] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel, "Differential-geometric newton method for the best rank-(r1, r2, r3) approximation of tensors," *Numerical Algorithms*, vol. 51, no. 2, pp. 179–194, 2009.

[126] P. Balouchian, M. Safaei, X. Cao, and H. Foroosh, "Unsupervised ranking of continuous emotions in images," in *Proc. British Machine Vision Conference (BMVC)*, 2019.

[127] C. Sun, I. Junejo, M. Tappen, and H. Foroosh, "Exploring sparseness and self-similarity for action recognition," *IEEE Transactions on Image Processing*, vol. 24, pp. 2488–2501, 2015.

[128] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034–3042.

[129] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[130] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[131] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[132] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 483–499.

[133] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[134] P. Balouchian, M. Safaei, and H. Foroosh, "LUCFER: A large-scale context-sensitive image dataset for deep learning of visual emotions," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1645–1654.

[135] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with rcnn," 2015.

[136] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[137] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE CVPR*, 2015.

[138] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE CVPR*, 2016.

[139] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," in *Proc. IEEE CVPR*, 2017.

[140] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *Journal of artificial intelligence research*, vol. 10, pp. 271–289, 1999.

[141] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *The Tenth IEEE International Conference on Computer Vision (ICCV'05)*, 2005.

[142] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proc. of ICPR*, 2004, pp. 32–36.

[143] L.-J. Li, H. Su, L. Fei-Fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Advances in neural information processing systems*, 2010, pp. 1378–1386.

[144] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3360–3367.

[145] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[146] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49–64, 1996.

[147] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. Siam, 1995, vol. 15.

[148] J. Gama, "Discriminant trees," *In practice*, vol. 1, p. 4, 1999.